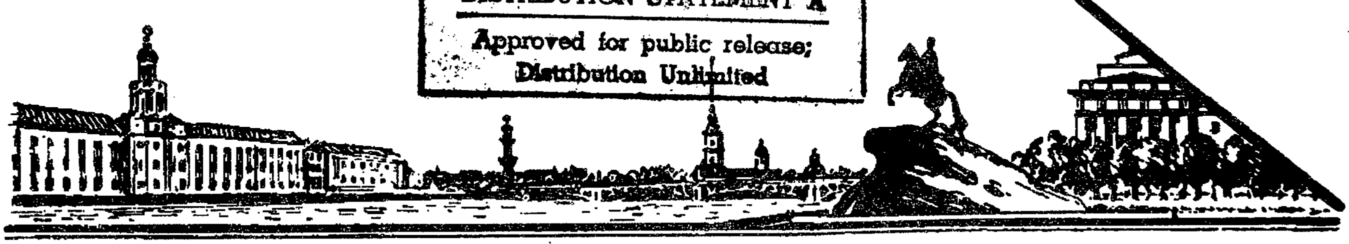
ICI&C'97

INTERNATIONAL
CONFERENCE ON
INFORMATICS
AND CONTROL

PROCEEDINGS

June 9-13, 1997
St. Petersburg
RUSSIA



ISBN 5-85733-078-5

PUBLISHED BY ST. PETERSBURG INSTITUTE FOR INFORMATICS AND AUTOMATION OF THE RUSSIAN ACADEMY OF SCIENCES (SPIIRAS)
39, 14 Linia, VO
St. Petersburg, 199178, Russia
spiiran@iias.spb.su

COVER DESIGNED BY NIKOLAY D. GORSKY

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
12 June 1997

3. REPORT TYPE AND DATES COVERED
Conference Proceedings

4. TITLE AND SUBTITLE
International Conference on Informatics and Control (ICI&C'97) Proceedings VOL 2

5. FUNDING NUMBERS
F6170897W0117

6. AUTHOR(S)
Conference Committee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
St.Petersburg Institute For Informatics & Automation
39, 14 Line, V.O.
St.Petersburg 199178
Russia

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
EOARD
PSC 802 BOX 14
FPO 09499-0200

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
CSP 97-1025

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 words)

The Final Proceedings for International Conference on Informatics and Control, 9 June 1997 - 13 June 1997

The Topics covered include: Intelligent control systems and information processing; Nueroinformatics and nueronet control; Robots' control and computer integrated technologies; Qualitative techniques for control and information processing tasks; Stochastic systems for information processing in motion control tasks; Safety control in complex systems, Transport Control (vehicles, vessels, aircraft, etc.); Satellite remote sensing in control and control related applications; Telecommunications: basics.

14. SUBJECT TERMS
Control, Control System, Advanced Computing, Computers, Electronic Devices, Instrumentation, Optimization, Signal Processing, Human Factors

15. NUMBER OF PAGES
419 - 848

16. PRICE CODE
N/A

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19, SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18
298-102

International Conference on Informatics and Control
(ICI&C'97)

PROCEEDINGS

VOLUME 2 OF 3

St. Petersburg, Russia

TABLE OF CONTENTS

CHAPTER V:

EXPERT SYSTEMS

THIS CHAPTER INCLUDES PAPERS

PRESENTED AT THE CONFERENCE SESSION:

EXPERT SYSTEMS

Organized by: ***Prof. Vyacheslav V. Ivanischev,***

Dr. Vladimir E. Marley.

USING EXPLICIT REPRESENTATION OF INFERENCE SEARCH METARULES FOR INFERENCE SEARCH CONTROL IN EXPERT SYSTEMS

Vladimir Tarasov
Department of Mathematics and Data Analysis,
Karelian Research Centre of the Russian Academy of Sciences
11 Pushkinskaya St.
Petrozavodsk, Russia, 185610
tarasov@centre.karelia.su

Abstract. This paper considers the use of metalevel tools for inference search control in expert systems. Offered here are tools for inference search control in the form of a system of language constructions, the basic ones of which are inference metarules, control predicates, control labels and inferences. These tools served as a basis for the development of an inference search control language designed for the description of query processing control in expert systems. This language allows for more precise description of the logic and techniques of an expert's reasoning about solving a particular task due to an extended set of control predicates, use of an explicit labels arrangement for backtracking organization, and description of solution search strategies in an explicit way with the help of inference search metarules. The accomplished program implementation of the inference search control language demonstrates possibilities of the proposed approach and can form a basis for development of appropriate tools. The designed language is approved in the problem domain connected with the search for function limits. A prototype of the knowledge-based system is constructed for solving training tasks of search for function limits.

Keywords. expert systems, inference search control, problem solving, decision support systems.

Introduction. One of the most promising trends in the development of computer systems is their intellectualization. This modern information technology is used in various areas, in particular, in decision support system creation [1]. The study and development of effective methodological, language and software facilities of building intellectualized computer systems is a topical problem. Nowadays development and improvement of a series of prototypes is a generally accepted approach to creation of expert systems (ES). This process entails an urgent and complex problem of constructing effective inference search engines most precisely simulating the logic and techniques of an expert's reasoning in a knowledge base.

The available inference search control facilities in expert systems are frequently based on general heuristic considerations or deductive inference methods, not reflecting completely the specificity of the tasks being solved, are of local flavour, are frequently rather poor and ineffective (e.g., the predicates *cut* and *fail* of the Prolog language, LEX-strategy of selection of the OPS5 language). Besides, general inference methods do not always fully reflect the features of an expert's reasoning in the given problem domain. As a consequence, the expert often has to adapt his reasoning to the used scheme of inference search control. And this scheme is not always convenient and effective which largely complicates verbalization of the expert's knowledge.

The presence of indicated disadvantages leads to the necessity of studying the question of creating more powerful and convenient language and software facilities with particular emphasis not on universality, but on convenience of tuning these tools to the considered problem class. From our viewpoint, such tools must help users (experts, programmers) to more precisely describe their reasoning logic when solving problems with special account taken of the features of the simulated problem domain, and also must facilitate modification and maintainability of the system. In this connection it is necessary to note the works by A. Bundy and B. Welham [2], T. Ishida et al [3].

As such a tool we offer an inference search control language (ICL) which allows for description of control of query processing in expert systems. The proposed language is based on metalevel facilities for inference search control. These facilities imply the possibility of defining features (predicates) on the current inference, performing operations on it, and they also contain means of interaction between the metalevel and the basic knowledge system.

An Approach to Inference Search Control Based on the Use of Inference Search Metarules. The use of metalevel facilities is one of the promising approaches to solving the problem of creation of inference search control means focusing on convenience of tuning these tools to the considered problem class. The proposed approach to inference search control using inference search metarules is based on the works by A.O. Slissenko [4,5]. Figure 1 shows the structural diagram describing this approach to inference search control.

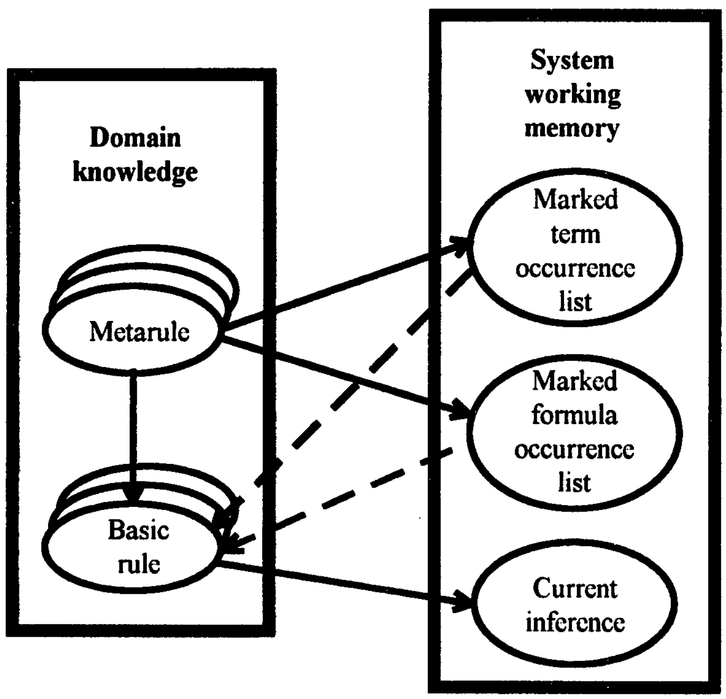


Fig. 1. Structural scheme of inference search control, based on explicit representation of inference search metarules

This approach involves dividing all the knowledge about the simulated problem domain into two levels: problem-oriented (domain knowledge) and inference search control (control knowledge). In this case, the control knowledge can be considered as metaknowledge in relation to the domain knowledge.

Let us introduce definitions of the following main notions: an inference and substitution.

An inference is a list of formulae supplied with additional information. Each formula F on the list is marked as an assumption or conclusion (both marks can be present simultaneously). In the case, when a formula is a conclusion, it has a reference to its origin, i.e. the name of the rule from which it was deduced, and a sublist of the formulae located to the left of F, from which it was obtained (when the formula has both marks there may be no reference to its origin).

A substitution is a rule of the form "in a term θ replace its (the i-th, any single or everyone) subterm τ by the term t under the precondition $F(\theta,\tau)$ and postcondition $G(\theta,\tau,t)$", where F and G are formulae of some language (e.g. an occurrence language or its fragment). A formula substitution is defined in the same way.

In our case the problem-oriented level (basic level) consists of a query language (basic language) and a set of inference rules (basic rules). The query language is designed for formulating specific queries to the system. The basic rules represent knowledge about the problem domain. In doing so, each basic rule validates one step of the simulated reasoning and is an algorithm transforming the formulae list representing the inference into an extension of this list.

All basic rules are divided into substitution-like rules and rules applying a special algorithm. A substitution may be formal or with value evaluation. A special algorithm implements such operations which cannot be described with substitutions, for example, operations on inferences.

The invocation condition of a basic rule is a sample for matching and is represented as a metaformula or metaterm describing appearance of the formula or term that is to be transformed. Formulae and terms for matching are extracted from the lists of marked occurrences of formulae and terms. In case of successful matching the necessary transformations are made and a new element is added to the current inference.

In our understanding, the inference search control level (metalevel) contains the methods and ways of search for solutions to problems (queries to the system), these being represented as inference search metarules. The metarules contain information on how to use the basic rules. Each metarule is an algorithm which according to the current inference produces the list of basic rules and those occurrences of formulae and terms to which it is necessary to apply the mentioned basic rules. In other words, a metarule describes some situation and those actions which are necessary to undertake when the given situation is recognized. The occurrences of the formulae and terms to be transformed are put on the lists of marked occurrences of formulae and terms.

The invocation conditions of metarules are represented as control formulae which are constructed from control predicates and logical connectives. The control predicates describe properties and features of inferences, formulae, terms and are divided into built-in ones and those defined by the user. The built-in control predicates are intended for checking the appearance and other properties of inferences, formulae, terms and for controlling the construction of the lists of marked occurrences of formulae and terms.

The inference search engine is formalized as a program consisting of a set of the metarules. The interpreter cycle-wise looks through the invocation conditions of metarules in that order in which they are put in the inference search program. The interpreter executes the first metarule which has a satisfied invocation condition. After that the control is transferred to the beginning of the next loop.

The exit condition and exceeding of the given limitation on the inference length are checked in the beginning of each next cycle. The exit condition is a control predicate defined by the user, its true value indicates that the answer to the query is found.

If the invocation condition of every metarule is not satisfied or the given length of the inference is exceeded on a next step, it means that the system cannot process the given query due to lack of the domain or control knowledge.

The interpreter operation can be formally described as follows.

```
L:=Q;
i:=0;
while(NOT ExitCond(L)  AND len(L)<n AND i<m )
begin
   i:=i+1;
   if(Ω1(Q,L,x)) then
   begin
      L:=apply(l1);
      continue; // transfer of control to the loop beginning
   end
   if(Ω2(Q,L,x)) then
   begin
      L:=apply(l2);
      continue;
   end
   . . .
   if(Ωk(Q,L,x)) then
   begin
      L:=apply(lk);
      continue;
   end
   HALT;
end,
```

where L - the current value of the inference under construction, x - any sublist of L, Q - the source query, Ω_i - a control formula, l_i - a list of basic rules, apply - the statement of applying basic rules, ExitCond - the exit condition.

Backtracking is organized with the help of control labels and metarules describing possible dead-end situations and methods of their resolving. Such operations on inferences as discarding dead-end branches or arranging control labels are executed by special basic rules which are built-in.

Inference Search Control Language. On the basis of the chosen approach we have developed an inference search control language (ICL), which is mainly designed for providing the user with convenient tools for description of algorithms of an inference search in his problem domain, and for quick selection of optimal ways of query processing at a stage of prototyping ES.

ICL is oriented at the description of symbol manipulations and includes control primitives and such classes of objects as domain constants, domain variables, functions, terms, predicates, formulae, inferences, metaconstants, metavariables, metafunctions, metaterms, metapredicates, metaformulae, control predicates, control formulae, basic inference rules, inference metarule, control labels, an inference search algorithm.

Domain constants, domain variables, functions, terms, predicates and formulae are used for formulating queries to ES. Metaterms and metaformulae designate terms or formulae, or set of terms or formulae. Metaterms and metaformulae are employed for describing appearance of terms or formulae in substitutions, and as arguments of control predicates.

Control predicates are designed for describing properties of ICL syntactical objects. Control predicates can denote features of terms, formulae and inferences. The user constructs control

predicates from built-in ones and Boolean operations. Control predicates are used for constructing control formulae.

Control formulae designate some statements about ICL syntactical objects and are used for constructing the IF parts of metarules and defining control predicates.

Basic rules are designed for representing statements about the problem domain, and describe those steps which are necessary to undertake to solve the task. An example of a basic rule applying a formal substitution follows.

BR43. Square difference. $a^2 - b^2 = (a-b)(a+b)$.

```
TSUBST(POL_SqrDif; subtr(pow(s,2),pow(t,2));
   mult(subtr(s,t),add(s,t)); PrepareTerm)
```

The invocation condition of a basic rule is unification of the sample for matching with the marked occurrence of a term (formula). The terms (formulae) to be probably transformed are transmitted through the lists of marked occurrences of terms and formulae, which are built during the check of the control condition of the corresponding metarule. In other words, the lists of marked occurrences of terms and formulae determine which occurrences of the term (formula) indicated in the sample for matching will be replaced or transformed.

Metarules are used for formalizing an inference search algorithm, which constitutes the main body of the inference search engine. Metarules describe concrete ways of realizing the selected method of problem solving. Each metarule describes some situation and those actions which are necessary to undertake in that particular situation.

The situation is characterized by a control formula, which makes it possible to formulate not only local conditions, for example, presence of some fact in the system working memory, but also more general conditions, for example, appearance of the current inference in the whole or presence of a dead-end situation. The actions described by the metarule are defined as a list of basic rules, some of which can be special rules used for the description of backtracking.

As an example we shall cite the metarule "Single Inference":

MR2. Single Inference. If the current inference consists of one formula and the expression falling under the limit sign and included in this formula of the current inference is product, difference or sum of functions, then mark the formula of the current inference with the label "single inference" and apply the basic rules about quotient, product, difference and sum of functions.

```
SingleInf :
   If((SingleInf(cur_inf)&ArithmOper(first_form)))
   Apply(label(single_inf),CL_FuncQuot,CL_FuncProd,CL_FuncDif,
      CL_FuncSum);
```

Control labels are designed for referencing the elements of the inference and are applied to organize backtracking. It is possible to mark with a control label any formula of the current inference and in further to backtrack to the marked formula. Using control labels the inference search algorithm can be precisely adjusted to the problem domain semantics due to selective backtracking only in those points where it is necessary.

Control primitives are employed for writing programs in the inference search control language. They are divided into built-in control predicates, carrying out construction of the lists of

marked occurrences of terms and formulae, and built-in basic rules, performing operations on inferences and control labels.

ICL Program Implementation. The ICL program implementation is a library of classes, objects and subroutines in the C++ language. The library contains abstract data types used for arranging inference search control in ES. The main classes included in the library are the following: basic rules, metarules, current inference, solving algorithm, control formulae, control predicates, control labels, terms, metaterms, formulae, metaformulae and inference engine. On the whole, the library includes 57 classes and 18 subroutines developed originally. The total volume of the source codes is 160K.

The library of the developed classes is an extension of the standard C++ language intended for working with lists of special sort, and is oriented to the Borland C++ compiler for MS Windows. The main module from the view point of the program implementation is "Inference Engine", which contains the implementation of all library classes, functions and predefined objects.

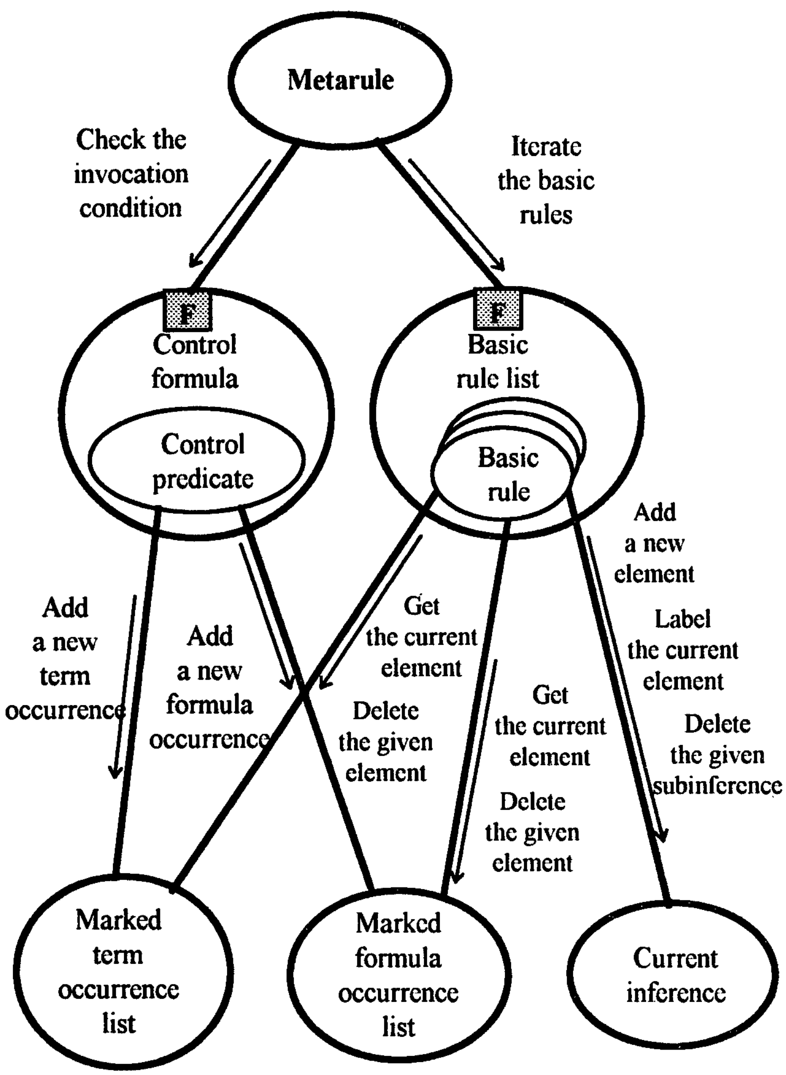


Fig. 2. Interaction of the main objects when building an inference

The main object is "Inference Engine". It has the structure and behaviour necessary for an inference search control. It includes as fields such objects as a solving algorithm, current inference, control labels list, lists of marked occurrences of terms and formulae. The methods of the "Inference Engine" class allow for initiation of query processing and input of an ICL-program at any moment.

The inference search itself is organized as interaction between objects of the following classes: metarules, basic rules, control predicates current inference, and lists of marked occurrences of terms and formulae. The diagram in Fig. 2 depicts the semantics of the key mechanism of the project – interaction of the main objects when building an inference. This diagram shows interconnections between the objects and the messages which they ex-

change with during the operation.

The program in the inference search control language is situated in a regular text file named by the user. The ICL-program includes the description of metaconstants, metavariables, metafunctions, and metapredicates, and also contains the definitions of control predicates and the solving algorithm – a list of metarules.

Practical Using ICL. The obtained results served as a basis for the development and implementation of a knowledge-based system for solving problems of search for function limits. The two-level architecture of knowledge representation was used for the system development.

The basic level includes knowledge accumulated in the theory of limits, and stated, for example, in the textbooks – common properties of limits, particular limits, rules of elementary function calculations and uncertainty removement, etc. This knowledge is formalized in the form of facts and basic rules which should compose a set sufficient for solution search. The basic level also comprises heuristics expressing informal knowledge used with the purpose of increasing the efficiency of solution search in the considered problem domain.

The metalevel contains the description of problem solving methods represented as metaknowledge. They include the description of ways of simplifying limits for expressions of various sort (rational, fraction-rational, trigonometrical etc.). The metaknowledge is represented in the form of metarules defining the selection of basic rules and ways of their use for solution search in the considered problem domain.

The characteristics of the ES prototype are: number of domain inference rules – 94, number of inference metarules – 25, average time of search for a function limit for a computer with the processor 486DX4 – 5 sec.

Conclusions. The accomplished research on the problem of development of inference search control facilities in expert systems has shown the possibility of practical realization of the idea of an inference search control based on extracting control knowledge from all the knowledge on the problem domain and on explicit defining this control knowledge as inference search metarules.

The proposed approach to an inference search control has the following distinctive features.

1. The developed system of the language constructions, which has been formalized as a specialized programming language, allows for description of query processing control in ES in the form of a program in ICL. This simplifies ES prototyping, and also its consequent modifications and maintenance.

2. It is possible to explicitly define features on the current inference and to perform operations on it, e.g. discarding a dead-end branch, and this facilitates tuning the inference search algorithm to the particular problem domain.

3. In the developed inference search control language none of the strategies of an inference search control is fixed beforehand. For an inference search control are used:
a) a rather large set of built-in predicates designed for controlling the search for a solution;
b) control predicates defined by the user and customized to the particular problem domain;

c) explicit arrangement of labels for executing backtracking only in those places where it is necessary;
d) inference search metarules explicitly specifying the strategy of search for a solution.

4. The control of search for a solution is described in the terms of syntactical objects of ICL, and rather general notion of an inference rule is used, which makes it possible to use the developed tool in problem domains with various semantics.

The inference search control language has been implemented in the C++ environment as software that can be used in the set of various tool complexes for building intellectualized computer systems. The question of expediency of application of this technology to creating production management computer systems is investigated nowadays [6]. Such a system alongside with traditional knowledge (e.g. mathematical models of production processes) can include subjective professional knowledge and experience of highly skilled experts about their ways of solving specific industrial tasks.

The obtained results served as a basis for the development and implementation of a knowledge-based system for solving problems of search for function limits. The experiments with the system have shown its efficiency and give possibility to hope for rather wide applicability of the proposed approach.

References

1. Dutta, A. (1996). Integrating AI and Optimization for Decision Support: A Survey. Decision Support Systems, N. 18, pp. 217-226.
2. Bundy, A., B. Welham (1981). Using Meta-level Inference for Selective Application of Multiple Rewrite Rule Sets in Algebraic Manipulation. Artificial Intelligence; N. 16, pp. 189-212.
3. Ishida, T., Y. Sasako, K. Nakata, Y. Fukuhura (1995). A Meta-Level Control Architecture for Production Systems. IEEE Transactions on Knowledge and Data Engineering, V. 7, N. 1, pp. 44-52.
4. Slissenko, A.O. (1991). On Measures of Information Quality of Knowledge Processing Systems. Information Sciences: An International Journal, V. 57-58, pp. 389-401.
5. Slissenko, A.O. (1990). Technological Environment for Expert Systems Development. Proceedings of the International IFIP Conference on Artificial Intelligence – Industrial Applications. Leningrad, Elsevier Science Publishers.
6. Vdovitsyn, V.T., V.V. Tarasov (1996). On Applying KBS-technology to a Pulp-and-Paper Mill Control Automation. New Information Technologies in Pulp and Paper Industry. II International Scientific and Technical Conference, Petrozavodsk, pp. 33-34.

ADAPTIVE IDENTIFICATION ALGORITHMS IN MARKETING STATISTICAL EXPERT SYSTEMS

N.N.Bakhtadze, V.A.Lototsky, E.M.Maximov
Institute of Control Sciences
65, Profsoyuznaya, Moscow, 117806, Russia
Tel. (095)3349201, e-mail: lotfone@ipu.rssi.ru

Abstract. The concept of statistical expert system for market state analysis and prediction and decision making support is given. New robust adaptive identification and prediction algorithms are suggested for use in statistical part of the system.

Keywords. Marketing, identification, expert system, robust adaptive algorithms.

Introduction

The analysis of demand for different products or services is primarily important both for pricing policy and optimal stock replenishment policy (both in quantity and assortment) determining . These are key elements of any corporation marketing strategy [1 - 3].

Last years demonstrate fast acceleration of the use of formal mathematical techniques in marketing investigations. E.g., in 1980 in USA regularly use mathematical methods only 25% of managers, and at least once used them 68%; in 1987 80% of marketing strategies of American corporations resulted from scientific research of complex segmented market [4].

The Russian businessmen interests in scientific marketing research are first of all concentrated in the area of demand prediction as background for pricing decision making [4, 5]. The western researchers experience illustrates the importance of combining mathematical (mostly, statistical) techniques and expert knowledge for effective demand prediction.

In this paper, we suggest an approach to design of marketing control (decision making support) system for a large-scale company, using, in particular, adaptive and robust identification and prediction algorithms.

A new class of such algorithms will be shortly described.

Main functions of Marketing Investigation, Optimization, and Prediction System (MIOPS)

The marketing system must have a number of functional abilities, main of which are as follows.

- The initial data base support and updating. DB must contain external and internal information. The external one is first of all as full as possible data about competitors, both statistical one (e.g., competitors prices), and qualitative one (e.g., about competitors marketing policies). Then, it may contain main currencies exchange rates; parliament, government, etc. actions influencing the market state; estimates of market quotes of different competitors, etc.

The internal one contains prices, sales, stocks, etc. statistical data; results of sociological investigations of real and potential customers, etc.

- Identification of market, and prediction of sales with respect to pricing policy and other relevant exogenous and endogenous variables.
- Expert analysis of marketing information, forecasts, etc., and adaptation of formal statistical algorithms to expert knowledge.
- Optimal inventory control, that is, order (for stock replenishment) determination, as a function of current stock level, prices, demand forecast, etc.
- Preparing different reports in form of graphics, tables, diagrams, histograms, etc.

The structure of marketing control system

The marketing control system must contain 4 main subsystems (Fig.1). The first one must collect outer and inner marketing information, support the corresponding data-base, make standard information pre-processing (averaging, sampling, normalizing, etc.), prepare reports, etc.

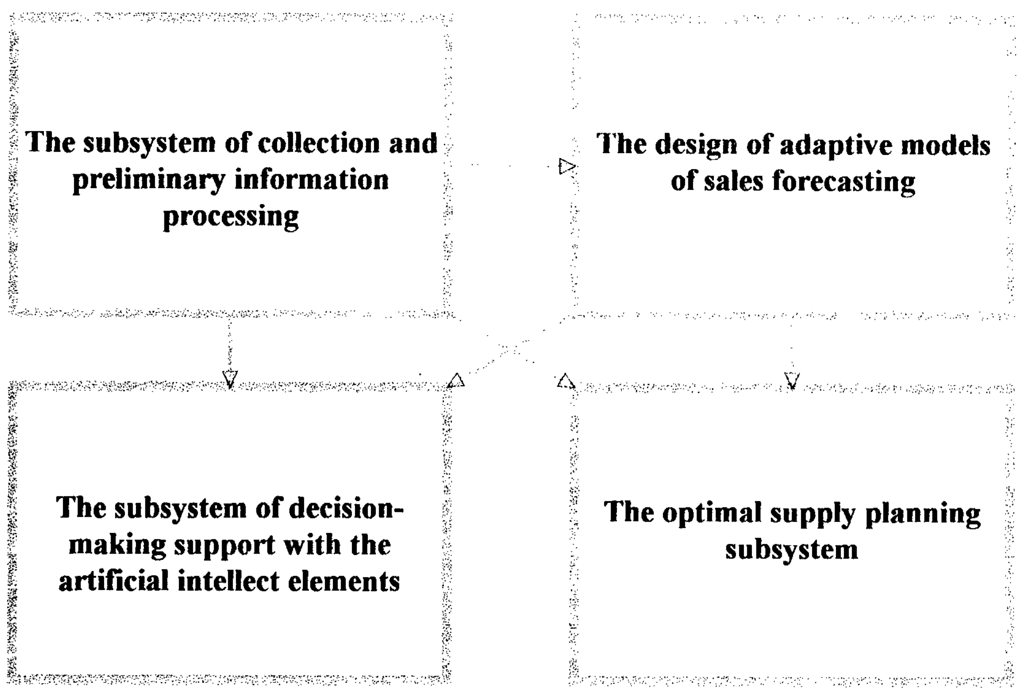


Fig.1

The second subsystem is a central analytical block of the system. Here the statistical data are processed to identify the market model, to predict sales and other characteristics of the company under consideration activity. To do it, this block must contain a number of algorithms. In MIOPS, both the «classic» techniques are used (e.g., recurrent least squares, adaptive algorithms from [6 - 8], etc.), and new robust adaptive algorithms, specially developed for

marketing research, which will be described further. Different algorithms are used to process statistical data, and the best in terms of prediction error is chosen.

The expert part of MIOPS support a real-time dialog with the expert, when he is asked about some qualitative factors, influencing the market (trends in competitors behavior, advertising efficiency, etc.) and choose his answer from several suggested alternatives in terms of «more», «much more», «less», «increasing», «slowly increasing», «stable», etc.

The sequential expert answers are quantitatively estimated, using Bayesian approach, and are used to correct forecasts. There is also possibility to adjust the parameters of algorithms in analytical block using expert knowledge.

The last block of MIOPS is dedicated for optimal stock replenishment and uses adaptive inventory control algorithms from [9, 10].

Robust adaptive identification and prediction algorithms for marketing research

The specifics of marketing problems is first of all rather high level of noises, incomplete apriori information, nonstationarity, correlated inputs. In this case, the following nonlinear randomized identification algorithm can be effective.

Let the identified object is as follows

$$y_k = h^T x_k + \varepsilon_k, \qquad (1)$$

where y_k -output, x_k - correlated input (both exogenous and endogenous) vector of dimension n, h - vector of unknown parameters, ε_k -noise, $k = 0,1,2,...$ - time. For the object (1), we use the following algorithm for calculating estimates sequence $\{\widetilde{C}_k\}$ of unknown parameters h in (1) from the measurements $\{y_k, x_k, k = 0,1,...\}$:

$$\widetilde{C}_k = \Xi_k C_k,$$

where

$$C_k = C_{k-1} + \frac{a}{n} M_k x_k \Xi_{k-1} \operatorname{sign}[y_k - C_{k-1}^T \Xi_{k-1} x_k],$$

$$M_k = \frac{k-1}{k}\left(M_{k-1} - \frac{M_{k-1} x_{k-1} x_{k-1}^T M_{k-1}^T}{k-1+x_k^T M_{k-1} x_k}\right),$$

$$M_0 = x_0 x_0^T + \rho^{-1} I,$$

(I - identity matrix),

$$\Xi_k = \operatorname{diag}(\xi_{k_i}), \qquad i = 1,\ldots,n,$$

$$\xi_{k_i} = \mathrm{sign}\Phi_{k_i}\,\mathrm{sign}C_{k_i},$$

$$\Phi_{k_i} = \begin{cases} \zeta_{k_i}, & \text{if } \zeta_{k_i} \neq 0, \\ C_{k_i}, & \text{if } \zeta_{k_i} = 0, \end{cases}$$

ζ_{k_i} – $(i = 1,\ldots,n)$ - the sequence of independent identically distributed random variables with simmetrical density function,

$\zeta_{k_i} \neq 0$, if $k < \infty$; $\zeta_{k_i} \xrightarrow[k\to\infty]{} 0$ almost surely with rate $1/k^2$;

$\hat{M}_k$ - the estimate of matrix $M = (M\{x_k x_k^T\})^{-1}$, and is calculated from the following equation

$$\hat{M}_k = \Big(\frac{1}{k}\sum_{i=1}^{k} x_i x_i^T + \frac{\rho}{k} I\Big)^{-1},$$

$$\rho = \text{const} > 0, \quad a = \text{const} > 0.$$

This algorithm is rather sophisticated (compared, say, to recursive least squares), but it has a number of advantages:

- it converges *almost surely;*
- it posesses *robustness* properties;
- the simulation results demonstrate good *convergence rate*;
- really, we have here *parametric class* of algorithms (where parameters are a, ρ, the distribution of ζ_n), which are applicable to various practical problems.

References

1. Sheth J.N.: *Marketing Theory: Evolution and Evaluation.* N.Y., N.Y., 1988.
2. Kotler Ph. *Marketing Essentials.* Copenhagen, 1984.
3. Vernon K.D.C. *Information Sources in Management and Business.* London, 1988.
4. Kolbasova A.B. Methods of demand analysis for new products (a review).*Economics and Mathem. Methods,* 1993, v.29, no. 1 (in Russian).
5. Hoskings A. *The Foundations of Business.* 1991.
6. Rajbman N.S., Chadeev V.M. *Identification of Industrial Processes.* North-Holland, Amsterdam, 1980.
7. Tsypkin Ya.Z. *The Foundations of Informational Identification Theory.* Whiley, N.Y.,N.Y.,1986.
8. Ljung L. *System Identification: Theory for the User.* Prentice Hall, Englewood Cliffs, N.J., 1987.
9. Lototsky V.A., Mandel A.S. *Models and Methods of Inventory Control.* Nauka, Moscow, 1991 (in Russian).
10. Bakhtadze N.N., Lototsky B.A. Fast convergent identification algorithms in inventory control problems. - In: Proc. of 10th IFAC Symp. SYSID'94, Copenhagen, 1994. v. 2.

THE INTEGRITY EVOLUTIONIZATION - NEW PARADIGM OF INTELLIGENT CONTROL TECHNOLOGIES

I.L. Bukatova
Russian Customs Academy, Scientific Information Center
Komsomolsky 4, Lybertsi-city, Moscow reg., 140009 Russia
tel: (7-095)503-76-11, fax: 5590336
e-mail: ilb208@ire216.msk.su

Abstract. *The modern researches on neuro-informatics are integrated in two main directions. First is conjugate with deepened and long researches in neuro-sciences with the purpose of subsequent imitation of found out mechanisms. Second - it is reduced to saturation neural networks by new functions and to connection additional mechanisms of organization of connections in networks. However in all these researches the evolutionary mechanisms on various levels of bio-organization as the mechanisms of synthesis of integrity with required properties are ignored. The evoinformatics is the set algorithm, programs and hardware, based on imitation of gears of natural evolution with the purpose of synthesis of structures, effective at processing of information in conditions of informative uncertainty. The evolutionary approach relies on further theoretical development of method of imitation not only genetic mechanisms of natural evolution, but also synergetic by integration of cooperative interaction of huge number of branches of artificial evolution. Given report submits the new conceptual platform of means of evoinformatics, including of trainees, adaptive and self-organizing evolutionary technologies. The directions of further researches, relying on last scientific results in this area [1-3] are scheduled.*

Keywords. Cognition, evolutionary processes, integrity, intelligent computer technology, knowledge, neurocomputer, self-organizing.

1. Soluble Problem.

Practically of any process of interaction of person with computer is directed on creation of information product, ensuring automatic (computer) execution of this or that function of some macrosystem or person as its element. Under macrosystem the wide spectrum of systems of various material realization and purposes is here understood: social, economic, natural (ecological), technical, hybrid (man-machine) systems.

In result of interaction of person with computer such difficult information systems as the global and local computer networks, expert systems, program complexes, specialized computer devices, neurocomputers, controlled data bases and knowledge and other information (computer) technology are created. The development of effective computer technologies as accumulation means of complex and activity knowledge is in present moment the most by priority task.

To such technologies the series of requirements and properties, is presented which are characteristic to intelligent abilities of person: the effective mastering and purchase of new knowledge, adaptability, learnability, selfcorrectability and self-organizing, maximal reformability under wide spectrum of tasks of applied area etc. In This connection the heaviest hopes are assigned on intelligent computer technologies [4].

Given report discusses the approach of integrity evolutionization as the conceptual platform of intelligent means of evoinformatics. The main outlines on uses of mechanisms of natural evolution in algorithm means of evoinformatics are tracked by author in work [5]. The

further detailing of used gears permits to allocate the spectrum of operators and parameters, affecting efficiency of evolutionary means (and technologies).

This spectrum of evolutionary "operators" is based on multialternative "Darwin's" imitation of mechanisms of evolution, processes realizing evolutionary of self-correction: the training, adaptation and self-organizing. The evolutionary processes of self-correction are classified as one-step-by-step (training) and multistep-by-step (adaptation and self-organizing), as single-stage (adaptation) and multi-stage (self-organizing) [6,7]. In turn the process of self-organizing (as well as other processes of self-correction) has many variants of realization, from which we note two extreme variants.

First is based on optimization, that is, allocation optimum (or suboptimal) branches of evolution, it is realized by software of evoinformatics [7,8]. Second variants is the mass-parallel calculations, in base of which the cooperation of branches of evolution and orientation on a neurocomputer means (and on hardware realization) [8,10]. However, for further development of means of evoinformatics the doubtless interest is presented the by processes of evolution, which work on evolutionary processes of self-correction. Clearly, that this following level of integration of gears and processes of natural evolution. On which to base and in which conditions the required integration is possible?

2. Integrity Evolutionary Representation.

As it is known [5,6-10], development of means of evoinformatics passes under thesis: to simulate the gears and processes of natural evolution on computer structures, to supply condition of occurrence of structure with required properties and functions of effective processing of information. Shall assume, that the artificial evolution of "population" of computer structures happens on the basis of integration of "Darwin's" mechanisms and mentioned evolutionary processes, that provides the integrity more general process.

Being oriented on intelligent means, naturally to consider the integration of evolutionary gears and processes at purchases of knowledge.

Thus, as main the following rules (situations) of integrity evolutionization [1] are considered:

1. The human-computer interaction is the knowledge acquisition process. It is realized in a system of cognition including the following components: an object of cognition, a subject of cognition, a image means of object in subject and a result of cognition.

2. The system of knowledge acquisition has a hierarchical structure of "nested" evolutionaring systems: every component is a complete system which includes the most level system as an element and which itself is an element of the most high level system. Beginning with the most high level this system has an object of cognition (macrosystem), a subject of cognition (man), means of object in subject image (technology and knowledge base), a result of cognition (knowledge base).

3. Integrity of a knowledge acquisition system, as its main property, is provided by integrity of the cognition process independently of the concrete material medium of any its element, the cognition process is the integrated interaction between macrosystem, man (user) and technology.

In this way the macrosystem represents a totality of nested systems with the correlated cognitive processes. The macrosystem knowledge is being formed by analogy. Its formation takes place because of two by two cognitive interaction of the adjacent systems (macrosystem-man, man-technology). The main sense of integrity evolutionization is an integrity keeping of the cognitive processes of the macrosystem and all nested systems during evolution.

Following these representations a structure of the cognition system is concretized (see on Fig.1). It includes a knowledge base of technology as an independent component.

However, both technology and its knowledge base makes up the integrated knowledge base of human-computer interaction. A man and his knowledge base are an integrated knowledge base of the macrosystem. A hierarchical nesting of components and cognitive processes, respectively, is the moment of principle in this representation.

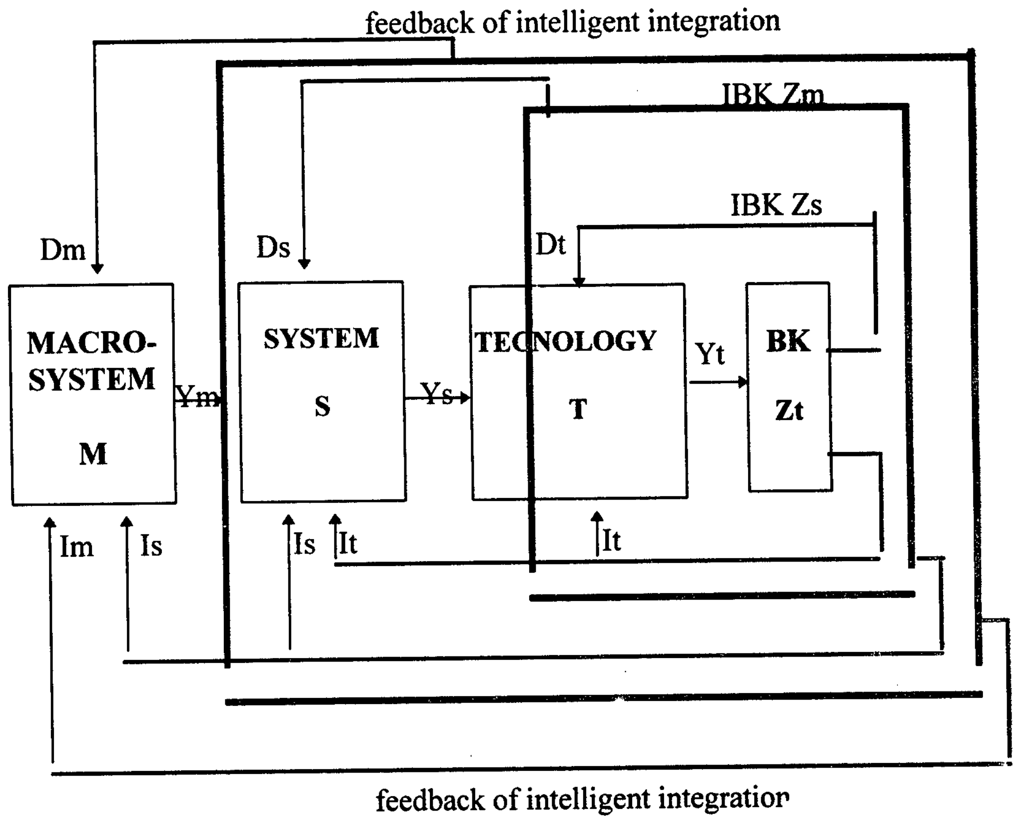


Fig.1. The structure of «nested» systems of knowledge acquisition process.

The processes of interaction between components singled out above are reflected by links of two types: intelligent and production interaction. Links of the first type determine the process of knowledge acquisition, links of the second type show the active interaction between structure components on a basis of the results of cognition. The direct intelligent links provide a system of the lowest level by information about a change of problem situation in a system of the highest level.

The back intelligent links give information about the prehistoric cognition which includes information about the changes in the adequate knowledge bases. If the whole complex of links is account for it is not difficult to concretize the processes of intellectualizing and intelligent integration within the limits of each nested system: technology, man, macrosystem. As a result, 29 cognitive process are determined.

They are classified as the following intelligent functions: intelligent correction, training, self-training, adaptation, mutual adaptation, self-organizing. The evolutionary processes connections which are realized with knowledge acquisition shown at Fig.2. As analysis shows, the concrete composition of intelligent functions in a process of knowledge acquisition of the macrosystem is determined by specific nature towards objective and subjective problem situations, prehistoric cognition, limits of a general process of knowledge acquisition or composition of realizing cognitive processes.

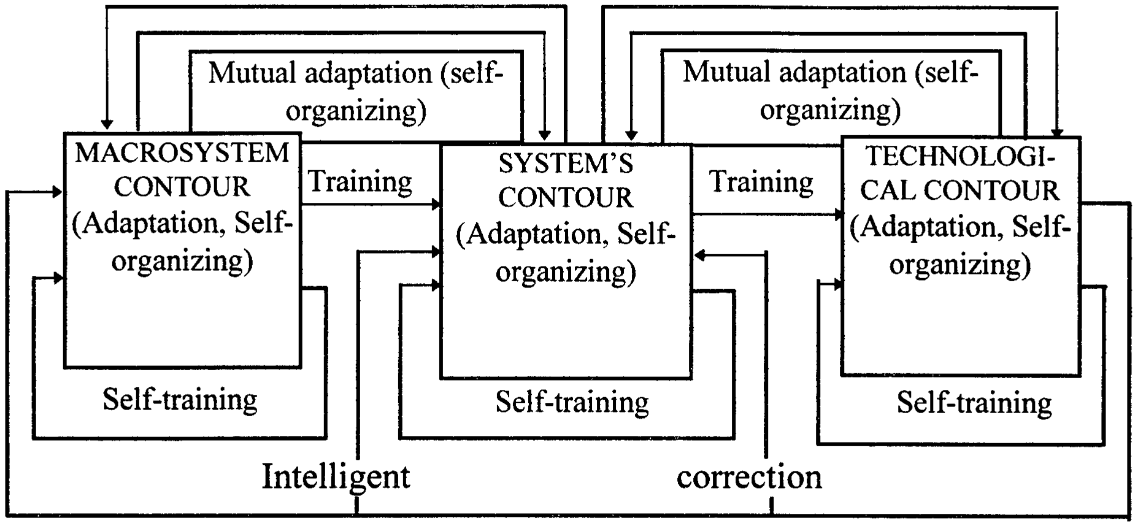


Fig. 2. Evolutionary processes of the macrosystem.

The analysis of man intelligent act shows his definite sequence in implementation of intelligent functions [2]. The analogous sequence exists in any contour of knowledge acquisition process. As a result, a all set of intelligent functions supported in the cognitive processes are divided into three subsets: initiating, realizing and completed functions. Conformity is set up for initiating and completed functions in a choice of any intelligent functions as an realizing that allows the cognitive process to be realized as a whole.

3. Intelligent Integrity Evolutionary Technologies.

The integrating result on base conducted in [1] of analysis are the common and private models of process of knowledge acquisition. In particular, the processes of direct interactions with object of cognition, self-cognition, mutual adaptation in contours and process of intelligent integration [1] are considered.

Common model reflects actually the technology of integrity evolutionization. It assumes the completion of following stages.

1. In conditions of prehistoric cognition the problem situation in the macrosystem is identificated as a totality of the problem situations arising in a application domain of macrosystem, man and technology.

2. The results of identification allows forming a solution of initialization of one of the cognitive processes and adequate intelligent functions for modernization of accumulated knowledge or acquisition of new knowledge, as well as correction of a knowledge system.

3. Within the limits of selected cognitive process, the adequate technological chain of knowledge acquisition is realized. Because of this, the problem situation is solved but the new one is formed that corresponds to operation execution of by p.1.

As a result the technology of integrity evolutionization determines the set and sequence of imitation evolutionary processes in various contours of evolutionaring systems: technology, system and macrosystem. Thus frameworks of each contour on the basis of evolutionary processes, adduced on Fig. 3, form the integrated knowledge in of appropriate base knowledge. Just in this moment on route, appropriate problem situation, is executed the integration of evolutionary processes (see the Fig. 2).

Thus, training (self-learning), adaptive (self-adaptive) and self-organizing evolutionary means, described in [5-10], realize only some part of possible combinations at integration of

evolutionary processes. That intelligent evolutionary technologies are realized only which include the evolutionary processes of technological contours.

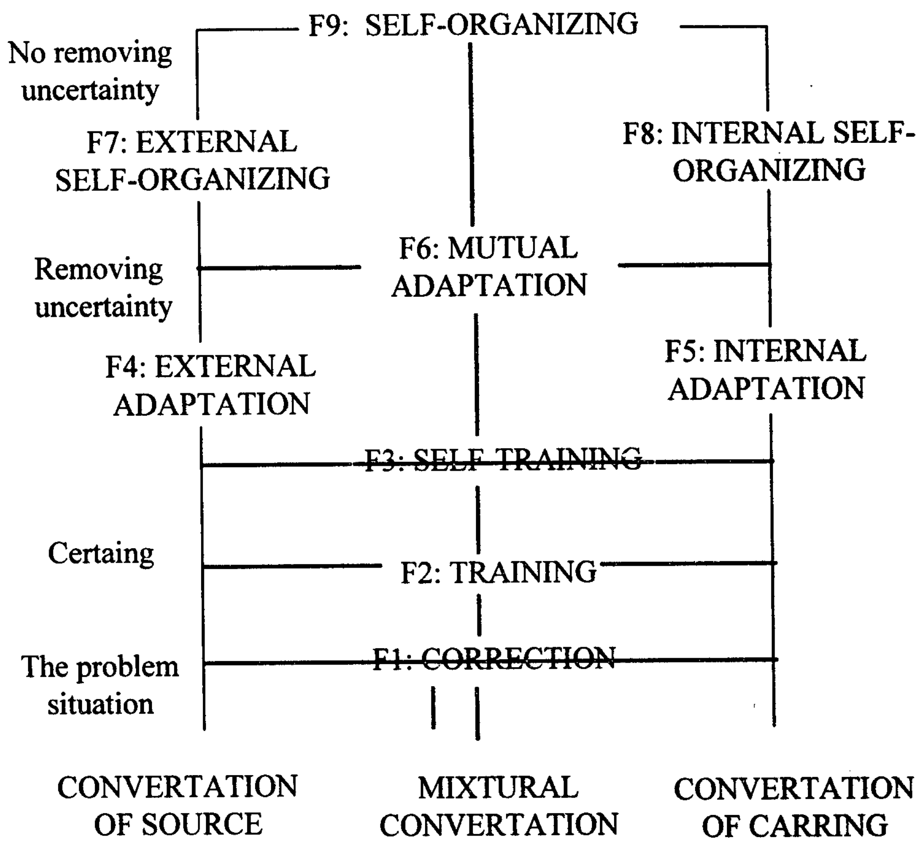


Fig.3. The structure of evolutionary processes of the knowledge acquisition.

Therefore the development whole of spectrum of intelligent evolutionary technologies is connected with further development of all combinations and interrelations of evolutionary processes, reflected on Fig. 2 and 3. In report these prospects are discussed on the basis of base concepts of cognitive dynamics, main its tasks and outlines their decisions [2,3].

Conclusions. The integrity-evolutionary approach to creation of intelligent computer technologies permits to decide the tasks of automation all greater numbers of stages of intelligent activity of person in conditions of evolution of its requirements. Thus other way of creation of high efficient means and systems of artificial intelligence, distinct from traditional, is displayed.

References

1. Bukatova, I.L., V.V. Makrusev (1994) The cognitive processes of evolutionaring systems. Preprint IRE Russian Academy of Sciences, 10 (598), Moscow. (In Russian).
2. Bukatova, I.L., V.V. Makrusev (1995) The cognitive dynamics: basic conceptions, main tasks and schemes of solution. Preprint IRE Russian Academy of Sciences, 8 (608), Moscow. (In Russian).
3. Bukatova, I.L., V.V. Makrusev (1995) The cognitive-productive metatechnology: constructive description. Preprint IRE Russian Academy of Sciences, 11 (611), Moscow. (In Russian).

4. Bukatova, I.L., V.V. Makrusev (1994) The intellectualizing of global information systems: base, concept, problem. Preprint IRE Russian Academy of Sciences, 7(595), Moscow. (In Russian)
5. Bukatova, I.L., Yu. V. Guliaev. (1991) From genetic algorithms to evolutionary computer. In Forrest (ed), Proceedings of the Fifth International Conference on Genetic Algorithms, pp.614-617, Urbana-Champaign, IL, Morgan Kaufmann, San Mateo, CA.
6. Bukatova, I.L. (1979) Evolutionary simulation and its applications. Nauka, Moscow. (In Russian).
7. Bukatova, I.L., Yu. I. Mikhasev, A.M. Sharov (1991) Evoinformatics: theory and practice of evolutionary simulation. Nauka, Moscow. (In Russian).
8. Bukatova, I.L. (1992) Evolutionary neurocomputer technology. Preprint IRE Russian Academy of Sciences, Moscow. (In Russian).
9. Bukatova, I.L. Evolutionary computer. Proceedings of the RNNS/IEEE Symposium on Neuroinformatics and Neurocomputers, v. I, pp. 467-477, Rostov-on-Don.
10.Bukatova, I.L. (1996) Self-training, adaptive and self-organizing evolutionary computations. Proceedings of First International Conference on Evolutionary Computation and Its Applications, Moscow, pp.169-182.

INTEGRITY EVOLUTIONARY INTELLECTUALIZATION OF INFORMATION-COMPUTER SYSTEMS

V.V. Makrusev
Russian Customs Academy, Scientific Information Center
Komsomolsky 4, Lybertsi-city, Moscow reg., 140009 Russia
tel: (7-095)503-7611, fax: 5590336

Abstract. *Theoretical research of increase efficiency problem of the traditional distributed control systems, which call later as information-computer systems (ICS) (global and local computer networks, human-computer complex, special computer device, computer and information technologies and etc.) are considered based on multi-level information streams coordination methodology and the intellectualizing process. In the result two main conceptual principles was obtained: 1. the evolution of the efficient technology is indeed the integration of evolutionary systems dynamics (technology, "nested" systems, macrosystem); 2. systems integrity of any level of nesting is defined from a knowledge acquisition process which is directed from the macrosystem to the technology. These principles had allowed to carried out the analyze of basic knowledge acquisition model means: the object of knowledge acquisition (macrosystems), the subject of knowledge acquisition (system), reflection means the object in the subject (knowledge base), activities subject means (technology).*

Keywords. Intellectualization, adaptation, cognitive interaction, computer technology, knowledge, information system, self-organizing.

The macrosystem, automation of which is performed on the employment of traditional global information computer systems, are distinguished by their unstable nature for practical purposes. The major problem of qualitative improvement of macrosystems is the problem of compensation of a dominating character of subjective knowledge in conditions of integrity evolutionary intellectualizing of ICS [1]. A solution of this problem is connected with the necessity of including intellectualized ICS into the macrosystem when providing the integrity of cognition process. The basis for its solution is an evolutionary approach [2] because it has been just the imitation of mechanisms of macrosystem evolution enables one to pose and adequately solve the intellectualizing problem of ICS.

The macrosystem represents a totality co-evolutionary systems (technology, nested systems, macrosystem) and corresponding cognitive processes. The macrosystem knowledge is being formed by analogy. Its formation takes place because of pair by pair intelligent interaction of the adjacent systems. The main sense of integrity evolutionary intellectualizing is an integrity keeping of the cognitive processes of the macrosystem in automation. This scientific trend is based on an integrity representation of intelligent interaction between systems and the integration's dynamic of the intelligent cognition functions.

The cognitive process expresses cognition technology [3]. The integrity cognitive representation is based on the following principles: nesting of objects, processes, cognition results and progressive evolution of knowledge in keeping of integrity of cognition process. It is expressed as an complete intelligent act (cognitive statics) as well as a process of knowledge development (cognitive dynamics). The integrity representation is characterized by a totality of the nested structures the elements of which are objects and result of cognition and the connections is the processes of interaction between these structures. Connections of intelligent interaction (determines a process of knowledge formation) and productive

interaction (expresses interaction between elements on a basis of results of cognition process) are presented.

The integrity of intelligent process of interaction between systems is reduced to integration of their intelligent possibilities. It is determined by the following conditions: interpretation of mutual simplicity, orientation, reflection, monitoring and completion. Essentially, this is the cognitive process forming the new knowledge (meta-knowledge). By functional nature, this is the monitoring (controlling) process of selection of effective evolutionary trends for a corresponding interaction space.

Any intellectualizing of ICS can and must exist if only they are under a process of development. This process includes evolution of space interaction between the nested systems and has the active nature. The main sense of such process is the control of change of technological platforms when going from a traditional system to adaptive or self-organizing ones. In order to solve this problem one can use a method of integration's dynamic of intelligent functions of cognition. This is a method of evolutionary synthesis of meta-knowledge based on the algorithm system of integrity training, adaptation, self-organizing with a wide choice of structural initial data. The selected direction including the evolutionary simulation helps to develop methods and means of ICS intellectualizing technology. The achieved level of technological potentialities must ensure the integrity evolution of intellectual potentialities of ICS with due regard for macrosystem evolutionary requirements.

References

1. Bukatova, I.L., V.V. Makrusev (1994) The cognitive processes of evolutionaring systems., Preprint IRE Russian Academy of Sciences, 10 (598), Moscow. (In Russian).
2. Bukatova, I.L., V.V. Makrusev (1995) The cognitive dynamics: basic conceptions, main tasks and schemes of solution. Preprint IRE Russian Academy of Sciences, 8 (608), Moscow. (In Russian).
3. Bukatova, I.L, V.V. Makrusev (1995) The cognitive-productive metatechnology: constructive description. Preprint IRE Russian Academy of Sciences, 11 (611), Moscow. (In Russian).

PROBABILITY INFERENCE IN EXPERT SYSTEM SHELL FOR NON-RELIABLE MEASUREMENTS

Elena Yu. Grazdantseva, Andrey G. Oleynik, Olga V. Oleynik, Alexander Ya. Fridman
Institute for Informatics and Mathematical Modelling of Technological Processes
Kola Science Centre, Russian Academy of Science
24 A, Fersman St.
Apatity, Murmansk reg., 184200 Russia
e-mail: oleynik@ksc-iimm.murmansk.su

Abstract. The article mainly focuses on one feature of the formerly developed expert system shell in probability operation mode, namely on the inference process under unreliable input information. Probability description of processes in the object under investigation is generalized to consider possible faults in measurements.

Keywords. Expert system shell, probability inference, non-reliable measurements.

The expert system shell (ESS) developed in [1 - 4] was intended to support all standard modes of expert system performance (data and rules maintenance, queries generation, knowledge base (KB) control, the expertize itself and its explanation). The ESS admits both determined and probability inference. Below we investigate mostly only one feature of the ESS in probability mode, namely the inference process under unreliable input information.

In the given ESS KB comprises a set of rules, each of them may have the following general format:

IF E_1 [& E_2...], [THEN T_1 [with $P(T_1 / *)$], [& T_2 [with $P(T_2 / *)$]...]]
ELSE I_1 [with $P(I_1 / *)$], [& I_2 [with $P(I_2 / *)$]...]], (1)

where: Ei, Tj, Ik are conditions of a logical type, each of them is defined on the list of values of one parameter or variable;

P (* / *) are conditional probabilities of consequences at the validity of their preconditions (are set for probability inference);

& denotes a logical connective "AND".

Any rule must have a part IF and at least one of the parts THEN or ELSE. Any part of a rule may consist of any number of logical conditions Ei, Tj, Ik.

The logical conditions may have the following form:

< name > < sign > < sublist_of_values (n) >
or (2)
[p1≤] P(< name > < sign > <sublist_of_values (n)>)[≤p2]

In expression (2):

< name **>** is a unique name of a parameter or a variable. Both parameters and variables are assigned with a limited list of possible values which is fixed at data input. Parameters and variables have values of symbolic type and numeric type, respectively. Thus, one may use mathematical formulas in the rules, if the formulas contain values of

variables. If a result of calculations represents a value of a variable, it is rounded up to the closest value choosen from the list of possible values for this variable;

< sublist_of_values (n) > is a list of n values taken from the list of values of the considered parameter/variable. Let N be the total number of values. Then always N > n.

Any variable is defined within $[V_1,V_2]$ and assigned with N values, its list of admissible values v_m is calculated automatically.

P (*) is a current probability of getting the argument within the given interval of probability [p1, p2];

admissible < signs > and the respective ways of processing the <sublist_of_values(n)> are shown below in Table 1.

Table 1 Processing of Signs in Expert Rules

Type of data	sign	n	The way to process conditions in a rule
Any (parameter / variable)	=	<N	The condition is true if the datum gets at least one of the values from < sublist_of_ values (n) > (connection of values on OR)
	≠	<N	The condition is true if the datum gets any value from < sublist_ of_ values (n) > except the specified one
Variable	∈	2	The condition is true if the datum gets any value from the specified interval [v_m,v_l]
	∉	2	The condition is true if the datum gets any value outside of the specified interval [v_m, v_l]
	≤	1	The condition is true if the datum gets any allowable value smaller or equal to the specified one
	≥	1	The condition is true if the datum gets any allowable value more or equal to the specified one

The shell modes of actuating and operation were described in [1,4].

On their position in the decision tree all data can be divided into three categories: goal data (or just goals) that occur only in parts THEN or/and ELSE of the rules from a project; leaf data (leaves) you can meet in parts IF only; other (intermediate) data can be found both in preconditions of one rule(s) and in consequences of other rule(s). A user can choose any goal or intermediate datum as the goal of every particular examination.
In any operation mode, before the beginning of an examination, the time of the last DB updating is compared with the time of the last DB control, and an alert signal appears if DB was changed after checking it.

Any condition of a kind (2) is considered as a fact if it is contained in consequences of a realized rule or entered in ES from the outside. To simplify the analysis, complex conditions (2) are automatically splitted (depending on their signs and list of allowable values) into simple facts of a kind as Aj= Aij. Then, serially for each parameter from the list of facts, the true consequences are generated and treated as facts on the next iteration of direct search. For more details on the inference algorithm refer to the section "Examination..." below.

Since each condition in consequences of rules contains different parameters, during the inference it is easy and convenient to split the rules of kind (1) into components of the following format:

$$\text{IF } A_j\text{, THEN } H_i \text{ [with } \tilde{P}(H_i/A_j)]. \qquad (3)$$

Before proceeding with processing of the rules of the kind (3) we shall consider two possible cases of setting the initial data for (3), particularly for probability inference (when the rules contain probabilities or/and unreliable measurements are possible). These data may be either calculated on the basis of the accumulated statistical information or directly input by an expert.

In the first case the data are set in the following shape:

$$P(H_i),\ i=1,2,...,n,\ \ \tilde{P}(A_j/H_i),\ i=1,2,...,n\ ,\ j=1,2,...,J\ ,$$

$$\sum_{i=1}^{n} P(H_i) = 1, \qquad \sum_{j=1}^{J} \tilde{P}(A_j/H_i) = 1 \qquad (4)$$

where the probability $\tilde{P}(A_j/H_i)$, i=1,2,..., n; j=1,2,..., J means the statistics received as the ratio of number of successful measurements to the total number of conducted measurements.

To use rule (3) hereinafter, it is necessary to pass from the probability $\tilde{P}(A_j/H_i)$ to the probability $\tilde{P}(H_i/A_j)$. We use the Bayes formula for this purpose since the norming limits (4) take place for this formula. Hence,

$$\tilde{P}(H_i/A_j) = \frac{P(H_i)\cdot\tilde{P}(A_j/H_i)}{\sum_{i=1}^{n} P(H_i)\cdot\tilde{P}(A_j/H_i)},$$

where the denominator equals to the complete probability P(Aj), j=1,2,...,J. Here

$$\sum_{i=1}^{n} \tilde{P}(H_i/A_j) = 1. \qquad (5)$$

In the second case the expert directly sets the conditional probability P(Hi/Aj) in a rule (3). If the validity of the left part of (3) is concluded by processing results of some measuring technique or a device, the possibility of random failures during a measurement is to be considered in general case. For this purpose, let us define the tool

probability Pi, which represents the probability of correct measurement for the applied technique (device) [1, 3].

Any measurement may produce the same answer in two cases: when some event has really taken place and the process of measurement has passed without failures, or when the event has not taken place and the failure in measurements has happened to produce this result. Then conditional probability, appropriate to the statistics $\widetilde{P}(H_i/A_j)$, can be calculated as follows:

$$\widetilde{P}(H_i/A_j) = P(Hi/Aj)\cdot Pi + \frac{1}{n-1}(1-P(Hi/Aj))(1-Pi).$$

Thus, norming conditions (5) are observed and boundary cases are fulfilled, namely:

If $Pi \to 1$, then $\widetilde{P}(H_i/A_j) \to P(Hi/Aj)$,

that is, conditional probabilities during measurements are not deformed when there are no failures;

If $Pi \to 0$, then $\widetilde{P}(H_i/A_j) \to \frac{1}{n-1}(1-P(Hi/Aj))$,

that is, the output probability of a measuring device tends to the average value of probabilities of all incorrect measurements (no correct measurements are feasible).

Now we proceed to processing of the set of rules (3). Let K rules testify to a hypothesis Hi. $\widetilde{P}(H_i/A_1), \widetilde{P}(H_i/A_2),..., \widetilde{P}(H_i/A_K)$ make the respective set of conditional probabilities and it is necessary to find out the probability of the hypothesis Hi provided that $A_1,A_2,...,A_k$ are taking place simultaneously, that is, it is necessary to calculate the probability $\widetilde{P}(H_i/\bigcap_{k=1}^{K} A_k)$.

To calculate this conditional probability, we will use the Bayes formula, since norming conditions (2), (4) are observed, and $A_1,A_2,...,A_k$ are independent according to assumption [1]. Thus, we receive:

$$\widetilde{P}(H_i/\bigcap_{k=1}^{K} A_k) = \frac{P(H_i)\cdot\widetilde{P}(\bigcap_{k=1}^{K} A_k/H_i)}{P(\bigcap_{k=1}^{K} A_k)} = \frac{P(H_i)\cdot\widetilde{P}(\bigcap_{k=1}^{K} A_k/H_i)}{\sum_{i=1}^{n} P(H_i)\cdot\widetilde{P}(\bigcap_{k=1}^{K} A_k/H_i)} = \frac{P(H_i)\prod_{k=1}^{K}\widetilde{P}(A_k/H_i)}{\sum_{i=1}^{n} P(H_i)\prod_{k=1}^{K}\widetilde{P}(A_k/H_i)} =$$

$$= \frac{\prod_{k=1}^{K}\widetilde{P}(H_i/A_k)}{P^{k-1}(H_i)\sum_{i=1}^{n}\frac{1}{P^{k-1}(H_i)}\prod_{k=1}^{K}\widetilde{P}(H_i/A_k)}. \qquad (6)$$

If a hypothesis Hi is not present in consequences of any rule, the current absolute probability P(Hi) substitutes its conditional probability in formula (6). The appropriate simplifications of Eq. (6) for this case are obvious.

Examination of a Single Object. User interface in this operation mode differs a little from the one in conventional expert systems. There are options to choose: the goal of examination from the data list of the given project (except leaf data); the necessity to execute an external starting program (returning facts about leaves) prior to the examination; the necessity to save results of the examination in a file; displaying the leaf data, about which ES is going to ask the user, in order to skip the data which values shall be used for the given examination. Besides, for a determined project it is possible to choose whether to finish examination at occurrence of the first fact about the given goal or to continue examination until the existing rules produce new facts. At probability inference all existing rules are always examined.

If a starting program performance has resulted in writing some facts into DB, the examination begins with direct search. Each iteration of the direct search consists of a cycle for the data attributed with facts. For each datum the list of facts is concluded, all rules producing true consequences from those facts are marked, these consequences are recorded into the DB of facts. Considering new facts, tool probabilities of data are calculated and their aposterior probabilities are recalculated (after Bayes formula). Thus, different facts testifying to a value of a datum are assumed to be independent. If some iteration does not produce any new facts (an "idle" iteration), and the goal of the examination is not achieved yet, the system switches to the mode of backward search. Here the sequence of choosing the leaf data to ask the user about is determined by analysing the frequency of occurence the conditions with this datum in the tree of decisions. After the user inputs a fact about the requested datum, ES proceeds to the iteration of direct search, etc. Having completed the examination, the system displays the results and, depending on the mode of result saving, either switches to the mode of explanation or writes the results into an output file.

Examination of an Array of Objects. Here the examination is executed by a consecutive application of the given set of rules to each object from the array of objects.

This operation mode assumes the following.

Prior to the begining of the examination, all the facts necessary for inference must be transferred into the system (by executing the start program or by direct recording in the DB containing facts). Therefore the examination includes a direct search only, an idle iteration is treated as a malfunction and results in completion of examination for the current object.

The results of examination are always saved in a file. This file is updated at each subsequent start of the system in the mode of examination. In addition to opportunities provided for examination of a single object, at the examination of an array of objects it is possible to use retrospective information on actual values of the current and other objects of examination. The explanation of the results of examination can be actuated for any choosen object after completion of examination of all objects by starting the system in the mode of explanation with the number of this object as a parameter.

Details of Inference. During expertizing, the set of rules retrieved from KB is processed considering the type of the project (determined or non-determined) and the user-defined goal of the expertize. When processing, the logical conditions in a rule are considered as combined with the connective "AND", that is:

- if all conditions of the part IF are true, the value "true" is assigned to all conditions contained in the part THEN (if one exists), and the value "false" is assigned to the conditions of the part ELSE (if one exists), which have not been assigned before;
- if at least one condition of the part IF is false and the part ELSE exists, the value "true" is assigned to all conditions contained in the part ELSE, and the value "false" is assigned to all the rest conditions of the part IF, which have not been assigned before.

If any true consequences exist, the given rule is considered as a realized one and is not analyzed in further examination.

In expertizing mode, the current iteration includes examination of all the rules that have true consequences for the whole list of unused facts. Therefore at probability inference a situation may occur when there are facts both for and against the truth of part IF of the rule. In such a case the rule is marked as a realized one, the truth of part IF is not determined, both part THEN and part ELSE are considered true (with respective conditional probabilities).

To increase the reliability of the conclusions, it is strongly recommended that only the consequences, which have the conditional probability more than 0.5, should be included into the rules. Enlisting all possible alternatives in the consequences of one rule is also extremely undesirable as it can reduce to zero the influence of all the rules preceding the given rule in the chain of inference.

When using intervals of values for variables (signs $\in$ and $\notin$ in the Table 1), the boundary values are included in the interval. In "probability" projects (where the probability inference is applied) every value of a datum must be attributed (manually or by choosing from a set of standard distribution laws) with an apriori probability of this value P_0.

Manual input (or calculating during an examination) the probability equal to 1 for one of the values of a datum causes stopping of further inference for this datum.

Besides, as it has been mentioned above, every datum may be attributed with a tool probability of its reliable measuring Pi. The input Pi=1 is treated as an indication of reliable measurements of datum values. The input Pi=0 is treated as an indication that the respective tool probability will be calculated during examination. Such an input is admitted for goal and intermediate data only.

When calculating, the tool probability of a consequence is assigned with a tool probability of the most reliable precondition of this consequence. If a precondition consists of several conditions (they are logically connected as AND), its tool probability is calculated as a product of tool probabilities of constituting conditions.

As the ES exploitation has shown, in some applications it is rather difficult for users to set the conditional probabilities P(THEN/IF) or P(ELSE/NOT IF) defining the truth of consequences THEN/ELSE provided that the precondition IF is true. The probabilities are needed for creating a set of rules (1). Usually it is much easier to calculate the

"inverse" conditional probabilities P(IF/THEN) and P(NOT IF/ELSE) using statistical data . Therefore ES stipulates automatic generation of rules, based on recalculation of statistical data. According to the accepted approach, rules are generated for rather authentic interrelations only (if conditional probabilities of consequences exceed 0.5).

Dictinctions between the ESS and Conventional Expert Systems. The developed expert system shell is specialized to investigate non-stationary spatial objects. Representation of a control object as an array of elements of the same type (objects of examination) makes a basis for ESS realization. The array of elements is set in the coordinate axes of the object.

Main features of the ESS are the following.
1. The operation mode with an array of testing objects under examination is realized; tools to access retrospective data about state vectors and results of examination for both the current and other elements are available, that is spatial-temporary interrelations in the rules are admissible.
2. Probability description of processes in the object under investigation is generalized: unreliable measurements are considered, automatic generation of the rules based on statistical data is allowed; it is possible to use functions of current values of data probability in parts IF of rules.

The pilot version of the ESS is tested in a mine of Khibiny Apatite Deposit.

References

1. Dremov S.N., M.M. Kagan, M.A. Legashov, O.V. Oleynik, A.Ya. Fridman (1995) Expert System to Perform Diagnosis of a Rock Massif at an Underground Mine . Regional Information Networks. V.1. Apatity,.pp. 77 - 83.
2. Oleynik A.G., A.Ya. Fridman (1995) Generation of a Data Base for Computer-Aided Simulation on the Basis of Conceptual Model. Regional Information Networks. V.1. Apatity,. pp. 47 - 57.
3. Oleynik A.G. (1994) Software System to Support Some Steps of Computer - Aided Simulation on the Basis of Conceptual Model. Synthesis of Systems for Computer - Aided Simulation. Apatity,. pp. 78 - 85.
4. Grazdantseva E. Yu., A.G. Oleynik, O.V. Oleynik, A. Ya. Fridman (1996) . Expert System Shell for State Diagnostics of Non- stationary Spatial Objects. Papers of Second International Conference on Information Technology in the Barents Region (BAR-IT'96), Apatity, 18 - 19 September.

THE TECHNOLOGY OF MODELING ON A BASIS OF VISUAL REPRESENTATION

Vyacheslav V. Ivanischev
St.Petersburg Institute for Informatics and Automation
Russian Academy of Sciences
39, 14 line, 199178, St.Petersburg, Russia
e-mail: iv@iias.spb.su

Abstract. The paper is an overview of the new technology of modeling that uses the visualized description of the modelled object. The technology has two-layers nature. The first layer is the layer of algorithmic networks, and the second layer contains the icon representation of models. The technology is supported by COGNITRON tool modeling system.

Keywords. Algorithmic network, modeling, representation of models, mathematical formalization, cognitive modeling, visualization.

Introduction. Currently modeling as a method of reality investigating is developing in two directions which are to some extent opposite to each other. First, the conventional direction is oriented to the professional expert in the field of modeling experienced in modern techniques of mathematical formalization, art of the system analysis, possessing a number of other skills including some specific features of his or her character. The second direction was given impetus by the paradigm of new information technologies; their origin is associated with the name of academician G.S.Pospelov. A new information technology with reference to modeling means the technology of model construction by a non-professional user without an intermediary. Up to now this long expression ("the new information technology of modeling") has not got its own name. Such concepts as "soft modeling", "cognitive modeling", "automation of modeling" and some other are some of suggested versions. In all these cases achievement of such development level of the human-computer interaction means the level where construction and practical use of models become widespread and popular.

Algorithmic networks layer. In recent years great hopes have been placed on visualization of representations during interaction with the end user. With reference to the problem of modeling it means visualization of well-known modeling formalisms; for example, Petri networks, or creating new formalisms, that can be represented in graphic form and require mathematical knowledge of high school level. One of such formalisms that got the name of algorithmic networks was proposed in the beginning of 80's [1]. Based on it a few versions of SAPFIR tool modeling system, significant number of models and packages of applied programs have been developed. The new impetus to the technology was given by the idea of modeling of the whole subject (domain) areas [2], that required systematization of all basic results on algorithmic networks.

Definition 1. The algorithmic network is oriented without loops to graph vertex $G\ (V,\ X)$, where arcs designate the model variable $x_i \in X, i = \overline{1,n}$, and vertexes designate functional

correlations $f_j \in F, j = \overline{1,m}$, that connect values of model variables at a short interval of time Δt that corresponds to the step of modeling.

As a rule real and Boolean variables are used in the network as model variables. In the case of numerical vectors and matrixes [3] expansion is admitted. For the scalar case the set F, for the recent version of the COGNITRON system includes the following elementary functions $f_j, j = \overline{1,m}$:

$$F = \{x + y, x - y, xy, x / y, min(x, y), max(x, y), Sinx, Cosx, arcSinx, arcCosx, e^x, x^y, \\ ln\, x, log\, x, Antx, x > y, x \geq y, x < y, x \leq y, x = y, x \neq y, x \vee y, x \& y, y = \bar{x}, x(t) = y(t-1)\} \quad (1)$$

The syntactic rules of construction of algorithmic networks from $f_j \in F, j = \overline{1,m}$ are established by the following rules:

Connectedness condition

Uniqueness condition (2)

Loops absence condition

In more details, the vertexes in a network are linked in a network by the same-name variables according to orientation of the arcs, forming a coherent graph. In the constructed network the calculated variable cannot be met more then once. The network should have no loops that do not contain operators (except the delay operator Δt that implements function $x(t)=y(t-1)$).

Definition 2. The algorithmic network, satisfying conditions (2) is referred to as a correctly constructed network (CCN).

The network variables structure is $X = X_P \cup X_C$, $X_P \cap X_C = \varnothing$, where X is the set of the modeling variable of the network; Xp is the subset of subject variables (i.e. describing the object); Xc is the subset of serving (not interpreted) variables which become necessary when orienting the network to solving a particular problem.

$X = Xin \cup X\,\mathrm{int}\,p \cup Xout$, where $Xin, X\,\mathrm{int}, Xout$ are respectively, subsets of an input, internal and output variables of the network. In a general case $X\,\mathrm{int} \cap Xout \neq \varnothing$.

Multiple modeling. Formal-logic model of subject (domain) area. Experience of use of algorithmic networks in the practice of modeling without an intermediary has shown that among the end users (non-mathematicians) no more than 30% are capable of learning to build algorithmic networks independently. This fact suggests that there exists a sort genetic predisposition in some people to solving problems reflecting real word. The main question is what to offer the other 70 % of users?

The answer, in our opinion, can be found in developing a technology of multiple modeling [2], suggesting creation of model bases for whole subject areas. The dialogue with such bases is carried out at the level of a "push-button" procedure to choose models from the base, directed into "a zone of merge", where by formal transformations a model complex is created. For the end user the only problem remains in structuring the terms offered by fragmentary models of the base.

The theoretical basis of such an approach is formed by a constructive formal-logic model of subject (domain) area Ξ as [4]:

$$M = < T, P, A, \Pi > \qquad (3)$$

where T is the set of base elements $f_j, j = \overline{1,m}$, which form the basis for modeling formalism used (in our case algorithmic networks).

P is the syntactic rules of construction of base elements $f_j, j = \overline{1,m}$ in correctly constructed models (in our case of a correctly constructed network CCN).

A is system of axioms, as a set of fragmentary models $m_s, s = \overline{1,k}$, that were obtained by structuring of the given subject (domain) area. The models $m_s, s = \overline{1,k}$ will form the base of fragmentary models of the subject area Ξ.

Π is semantic rules (procedure) allowing to build the new syntactic correct sets on the basis of axiom from A.

At the present stage rules $p_C \in \Pi, c = \overline{1,l}$ are of interest as the procedures on the set of models $m_s, s = \overline{1,k}$, forming syntactically and semantically correct designs capable to fulfilling the system of axioms A. Some results in this area are given in [5,6].

The idea of multiple modeling is implemented on the example of the base of fragmentary models for the subject area "Farms in Russia".

References

1. Иванищев В.В. Автоматизация моделирования потоковых систем. Л.: Наука, 1986, 142 с.

2. Иванищев В.В. Технология множественного моделирования на основе формализма алгоритмических сетей.– В кн.: Сборник научных трудов КИИ-96 в трех томах. Том III. Казань 1996

3. Михайлов В.В. Матричное расширение языка алгоритмических сетей.–В кн.: Алгоритмическое моделирование: инструментальные средства и модели.–Л., ЛИИАН, 1992.-205 с.

4. Поспелов Д.А. Ситуационное управление: теория и практика.–М.: Наука.– Гл. ред. Физ.-мат. мет., 1986.–288 с.

5. Иванищев В.В. Когнитивный анализ процесса построения алгоритмических сетей.– В кн.: Алгоритмическое моделирование: инструментальные средства и модели.–Л.: ЛИИАН, 1992.– 205 с

6. Иванищев В.В., Михайлов В.В. Проблемы объединения фрагментарных моделей. V Санкт-Петербургская международная конференция "Региональная информатика–96". Тезисы докладов. Санкт-Петербург, 1996.

OPERATIONS OVER ALGORITHMIC NETWORKS.

V.E.Marley
SPIIRAS
39, 14 linia, VO
St.Petersburg, Russia
e-mail: iv@iias.spb.su

Abstract. The processes of creation and using of algorithmic models needs in system of operation for transforming the structure of models. The structure of models usually represented as algorithmic networks. Algebra of operation over algorithmic networks and the properties of operation are been considered.

Keywords. Algorithmic networks, nodes, links, sets, operation, intersection, union, difference, partial inversion, transformation of the description of node, aggregation desegregation, allocation subgraph.

The history of algorithmic networks and their use at creation of algorithmic models totals more than 15 years [1], there were about a hundred of publications, on various problems of theory and application of algorithmic networks (AN). Accumulated experience shows, that AN can be considered as some global formalism for representation of processes. In some of cases it is more strong then such widely known global formalisms as Pethry networks, parallel graph-schemes and etc. However, despite separate attempts [2-8] complete AN properties as of mathematical objects were not considered. The purpose of this paper is development of system of operations over AN, result of which are other AN and their properties, since standard operations over the graph applied to AN do not guarantee creation new AN.

Usually, in available publications, AN were defined as final directed graph, nodes are operators, and links are variables. AN is representation of algorithmic models and mappings some computation circuit, on which computing experiments over models are executed [3,5]. As AN is graph and representation of some algorithm, it is more convenient to define them as follows:

$AN ::= < P, Q, X, F, P \to F, Q \to X >$

Where:
If the network is not empty, all sets are not empty:
P - Set of nodes $\{p_i\}$;
Q - Set of links $\{q_k\}$;
X - Set of variable $\{x_j\}$;
F - Set of the operators $\{f_i\}$, are described as
$f_i ::= < in(f_i) > < f > < out(f_i) >$
Where:
f - Symbol of operation or function, i - index (number) of operation in the computing circuit, in a general case it can be name of some program module or other AN, the operators can be determined over real numbers, logic variable, scalars, vectors, matrixes, structures and etc.;
$in(f_i)$ - set of entrance variable (arguments, input) of operator;
$out(f_i)$ - set of calculating variable (results, output) of operator;
$P \to F$ - mapping P on F (one-to-one, isomorphism);

$Q \rightarrow X$ - mapping Q on X.

Only one variable corresponds to each link, variable can correspond a few links, $out(f_i)$ of all nodes AN is not intersected, AN does not contain contours, if contour do not include the node with operation of delay. The operation of delay serve in nodes AN for the task of an initial condition of simulated process and for the description of transition of simulated process through a step of modelling.

The link q_k in AN means, that for the operators appropriate nodes p_i and p_s are $\exists(x_j \in X) \in in(f_s) \cap out(f_i) \neq \varnothing$ and x_j is image q_k in $Q \rightarrow X$. We shall enter:
$out(AN) = \cup out(f_i)$ - set of all calculated variable AN,
$in(AN) = \cup in(f_i) \backslash out(AN)$ - set of entrance variable AN, $X = in(AN) \cup out(AN)$.

The isomorphism $P \rightarrow F$ makes possible to represented AN by only one set F. We shall speak about **equality two AN** ($AN_1 = AN_2$), if their sets F is equal ($F_1 = F_2$). We shall speak about **equivalence (hard) AN** ($AN_1 \sim AN_2$), if it is possible to define one-to-one mapping for their sets F ($F_1 \sim F_2$), if the elements put in conformity of sets F_1, F_2 have identical symbols of operations or functions f and capacity of sets $in(f_i)$ and $out(f_i)$ of this nodes are equal.

At first we shall define operations possible for two AN, consisting of one node with connected to it links or individual AN, thus we shall impose on operation some restrictions ensuring, that in result them will be received just AN.

Binary operations:

1) **Intersection** $AN_1 \cap AN_2 \neq \varnothing$, if and only if individual AN equal, $AN_1 \cap AN_2 = AN_1 = AN_2$, that one-to-one corresponds $F_1 \cap F_2$;

2) **Union** is $AN_1 \cup AN_2, \neq \varnothing$ in such cases:
- if the intersection AN is not empty and is equal the intersection;
- if the intersection of sets variable initial AN is empty ($X_1 \cap X_2 = \varnothing$), in this case AN consists from two components of connectivity submitted initial AN;
- if the intersection of sets calculating variable initial AN is empty ($out(f_1) \cap out(f_2) = \varnothing$), the intersection of set entrance variable both AN is not empty ($in(f_1) \cap in(f_2) \neq \varnothing$), and intersection entrance and calculating sets AN in any combination is empty, result is new AN which consists from two nodes, at which entrance links appropriated to intersection of sets entrance variable have a common beginning;
- if the intersection of sets calculating variable initial AN is empty ($out(f_1) \cap out(f_2) = \varnothing$) and the intersection of set calculating variable one AN with entrance set second AN is not empty, and for second AN is empty (for example, $in(f_2) \cap out(f_1) \neq \varnothing$, and $in(f_1) \cap out(f_2) = \varnothing$), result is new connected AN without contours consisting from two nodes connected among themselves links appropriate variable, on which the sets variable were intersected;
- if the intersection of sets calculating variable initial AN is empty ($out(f_1) \cap out(f_2) = \varnothing$) and the intersection of set calculating variable one AN with entrance set second AN for both AN is not empty ($in(f_2) \cap out(f_1) \neq \varnothing$ and $in(f_1) \cap out(f_2) \neq \varnothing$) and if one of nodes is the operator of a type "delay", result is new connected AN with a contour consisting from two nodes connected among themselves links appropriate variable, on which the sets variable were crossed;

in any of given above cases the union AN is defined by $F1 \cup F2$ under condition of fulfilment of restrictions;

3) **Difference** $AN_1 \backslash AN_2$ is equalled $\varnothing$, if $AN_1=AN_2$, or AN1, if $AN_1 \neq AN_2$, that corresponds $F_1 \backslash F_2$.

According to given definitions of operations over AN all of them are reduced to similar operations of sets theory over sets F_i.

Unary operations:

1) **The partial inversion**, is possible for individual AN - $Ob_{\{xj\}}(AN)$, if in node is the operator appropriate to the symmetric relation [5] on some subset variable connected with node, result is new $AN'=Ob_{\{xj\}}(AN)$ with changed structure in(AN) and out(AN) ($\{x_j\}$ - set variable initial AN (both entrance and calculating) forming prospective in(AN'). If variable of set $\{x_j\}$ is entrance, it remains entrance, calculating - inversions and becomes entrance for new AN'. Set X in resulting and initial AN is equal (i.e. $in(AN) \cup out(AN)=in(AN') \cup out(AN')=X$) The following situations are possible:
- $\{x_j\}$ determines only one operator appropriate the symmetric relation given by the initial operator - inversion possible ;
- $\{x_j\}$ a set does not determines any of the possible operators - is insufficient, the inversion is impossible;
- $\{x_j\}$ enables to determine at once a few operators appropriate to the symmetric relation - set is superfluous, the inversion is impossible.

The task of entrance sets, determining the operators, can come true:
- by given the list of all possible entrance sets and appropriate operators;
- by given the law to determine an opportunity of the inversion of the initial operator and kind of the new operator (for example [3,5,8], if in initial AN there was the operator a+b=d, any entrance set consisting from two variable connected by initial operator will determine allowable, and the kind of the operator will be determined by presence or absence in an entrance set variable d, if it is not present, the sum, if is present, a difference, to be calculated will be third variable not entering in an entrance set).

In particular, if set $\{x_j\}$= in(AN), the inversion is not present and received AN' is equal initial with same in (AN) and out(AN), if $\{x_j\}=\varnothing$, the set is insufficient.

2) **Transformation of the description of node** - $Ch^q{}_{pi}(AN)$ is possible, if mathematical expression describing the operator in node allows equivalent transformations not reduced to the partial inversion AN (as reduction similar, removal for brackets, uncover of brackets and etc.). Result is new $AN'=Ch^q{}_{pi}(AN)$ with other expression describing the operator f, and, it is possible with other sets in(AN), out(AN), p_i is transformed node (in this case AN), q is description of made transformation. It is possible, that in result of transformation the initial expression is broken on a few, designating intermediate results, in this case are in(AN)=in(AN'), $out(AN) \subseteq out(AN')$, to calculating links are added new appropriate entered intermediate resulting variable. If in result of transformation there was the union of expressions (by substitution), then eliminate absorbed intermediate variable, in(AN)=in(AN'), $out(AN) \supseteq out(AN')$, calculating links appropriate absorbed variable leave. It is possible, that transformations over expressions, describing the

operator were not made, then we have empty operation of transformation, designated $Ch^{\varnothing}_{pi}(AN)$, result of such operation will be initial AN.

Each of initial individual AN will be subnetwork AN received in result of their union. Links, on which initial AN incorporate in resulting, are named **internal**, the all other links are named **external.**

Subgraph of some initial AN is AN, all nodes of which are also nodes initial AN (if AN initial, AN′ subgraph, $F \supseteq F'$), empty AN is subnetwork any AN.

We shall define operations over complex AN.

Binary operations:

1) **Intersection** $AN_1 \cap AN_2$, the result is subgraph consisting from equal for given graphs nodes, i.e. can be determined on the basis $F_1 \cap F_2$;

2) **Union** $AN_1 \cup AN_2$, is equal to intersection AN and all of other nodes first and second AN, if all conditions similar above mentioned for individual networks are fulfil (the intersection of sets calculating variable for nodes initial AN is empty, if them is not included in intersection AN, and the linkage of nodes initial AN does not create contours without of node with the operator "delay". In case of fulfilment of conditions the operation of union can be given as $F_1 \cup F_2$.

3) **Difference** $AN_1 \backslash AN_2$, is equal first AN without subgraph appropriate to intersection initial AN, the operation can be given as $F_1 \backslash F_2$. If second AN was subgraph first ($AN_1 \supseteq AN_2$), the result of operation refers to as by addition of such network in initial greater.

All considered above operations over AN are reduced to similar operations of the sets theory over sets F_i.

Unary operations:

1) **Aggregation** $A_{\{pi\}}(AN)$ is the replacement of some connective subgraph AN by one node containing the operator, in which expressions appropriate to all operators of aggregating subgraph are incorporated, the external links of such node corresponds to external links of aggregating subgraph, and also calculating links, appropriated intermediate settlement variable of aggregating subgraph. In received node expressions calculating variable for all calculating links of aggregating subgraph should contain. $\{p_i\}$ is set of nodes of aggregating subgraph, if it is empty or $\{p_i\}$ has only one element, in result of operation the initial network remains without changes;

2) **Desegregation** $DA_{pi}(AN)$ is the inversion of operation aggregation, corresponds to replacement of some node AN subgraph constructed on the basis of a set of expressions determining the operator attributed to node, the set of external links such subgraph will correspond to set of links of node. p_i is desegregating node, if it is not given, or cannot be desegregated the result of operation is the initial network;

3) **Partial inversion** $AN'=Ob_{\{xj\}}(AN)$, result of operation is new AN, in which over separate the partial inversion is executed, the operation is possible if in AN there are the nodes, which can be inverted as it was described above for individual AN and if the

resulting of inversion give new AN. It realises by the algorithm of planning of calculation AN [3,5,8]. The algorithm of planning of calculation AN consists in consecutive looking of nodes AN and determination of an opportunity of partial inversion of each node (as in case of individual AN), mark of those nodes for which the operation of partial inversion already is executed and variables, which already can be calculated, and recurrence of looking with fulfilment of the specified actions until the operation over whole AN will be completed. The following variants of result of operation are possible:
- set is sufficient, there was the complete partial inversion initial AN, i.e. capacity of set F for initial AN is equal to capacity of set F′ for received network $|F|=|F'|$, $X=X'$;
- the set is insufficient, there was not the complete partial inversion, i.e. there was the inversion only subgraph of initial AN, $|F|>|F'|$, $X\supseteq X'$; if a set does not allow to determine the operator even for one node AN, the inversion is impossible and this set also insufficient;
- the set is superfluous, the inversion is impossible (it is determined at attempt the inversion of the next node).

If $\{x_j\}$ corresponds in(AN) for initial AN, result of operation will be same AN.

4) **Transformation of expression in node** - Ch^q_{pi} (AN), is always executed only for one node, completely corresponds to operation for individual networks;

5) **Allocation subgraph** on given subsets entrance and calculating for it links $AN'=S^O_I(AN)$, where O - set variable initial AN, all appropriate which links will be calculating links received subgraph, I - set variable initial AN, all appropriate which links will be entrance links received subgraph, $F'\subseteq F$, $X'\subseteq X$, $I\subseteq in(AN')$, $O\subseteq out(AN')$.

The following cases are possible:
- set O is given only, I is empty - received subgraph will consist of all nodes of calculating these variable and all nodes, from which it is possible to find a way comes to nodes calculating these variable, the process is finished by reaching nodes with the operators of "delay" or nodes, all entrance links of which have not initial node, union of all ways have such beginnings and ends gives resulting subgraph;
- set I is given only, O is empty - received subgraph will consist of all nodes, for which these variable are entrance and all other nodes arrivable of mentioned, the process is finished by reaching nodes with the operators of a type "delay" or nodes, all calculating links of which have not final node, union of all ways have such beginnings and ends gives resulting subgraph;
- if both sets is not empty, resulting subgraph is a intersection subgraphs received only for inputs and only for outputs (if the intersection is empty, resulting subgraph is empty);
- if both sets are empty, resulting subgraph is equal empty.

It is possible, when resulting subgraph will consist of some components of connectivity.

The properties of binary operations follow from accepted equivalence of operations over networks and operations similar to them over sets F, therefore the separate proofs of these properties are not brought, as it is wellknown properties and relations of operations over sets already repeatedly considered and proved in the sets theory.

On the basis of properties of operations over sets the operations of union, intersection and the differences of algorithmic networks are associative and distributive from each other. Operations of intersection and union are commutative and idempotentive. $\varnothing$ is

unit for union, zero for intersection, the right unit and left zero for a difference. If to consider, that all AN is subnetworks some general universal AN, such AN will be unit for intersection, zero for union. Additions any AN in AN, has properties of involution, submits to a rule of de Morgan.

Each of binary operations, at given initial data, gives one and only one result appropriate by this initial data (is proceeded from properties of the appropriate operations over sets).

The properties of unary operations over AN also are in many respects determined by transformations of set F.

The operations of aggregation and desegregation, can be defined as a sequence of operations of difference and union, thus deducted subgraphs are unequivocally given by entrance conditions of operations, and added subgraph is unequivocally given by definition of operation, the results of each of operations of a sequence are unequivocally determined, thus the results of operations aggregation and desegregation are unequivocally determined.

Operation aggregation, desegregation and the transformations of node set the relation of **weak (on sold function) equivalence AN** for given sets entrance and calculating variable (we shall it to designate as $AN_1 \approx AN_2$), i.e. received in result of these operations the network is equivalent initial one, because there is the sequence of operations aggregation, desegregation and transformation of the node applied to a received network, result of which will be initial AN.

Executing over AN operations of aggregation, desegregation and transformation of the node is changed a network topology, therefore it is necessary to specify conditions, at which concept of weak equivalence is considered:
- on initial AN some set $Y \subseteq out(AN)$ is allocated which is constant in result of fulfilment any of considered operations and in resulting AN′, about Y is considered equivalence on functions calculating variable of this set;
- $in(AN) \cap in(AN') \neq \varnothing$.

The operation of the partial inversion $AN' = Ob_{\{xj\}}(AN)$ is unequivocal, i.e. $\{x_j\}$, in case of fulfilment of operation at the complete or incomplete partial inversion, sets one and only one AN.

The operations of partial inversion, in case of end of operation by the complete partial inversion initial AN are set the relation of equivalence AN (we shall it designate as $AN_1 \equiv AN_2$), i.e. those AN, which can be received by operation of the partial inversion from some initial at its complete partial inversion.

Any sequence of operations of the partial inversion over some initial AN (in case of the complete partial inversion at each stage of a sequence) can be replaced by one operation of the partial inversion with final variant {xj}.

For operation of the partial inversion the also following statements are fair.
Let $AN' = Ob_{\{xj\}}(AN)$ and $\{x_j\} \subseteq in\ (AN)$, then $AN' \subseteq AN$.
Let $AN' = Ob_{\{xj1\}}(AN)$, $AN'' = Ob_{\{xj2\}}(AN)$, $\{x_{j1}\}$ ($\{x_{j2}\}$, then $AN' \subseteq AN''$.

The operation of allocation subgraph on AN $AN'=S^{O}_{I}(AN)$ is unequivocal, i.e. its result is unequivocally set by sets O and I.

If $I_1 \subseteq I_2$, $O_1 \subseteq O_2$, then $S^{O1}_{I1}(AN) \subseteq S^{O2}_{I2}(AN)$.

The considered operations set **algebra AN** and create a basis for the description of all possible actions, made over AN during creation of models, realisations of computing experiments and acceptance of the decisions.

The set F is as real the list of the descriptions of nodes, which is easily interpreted as records of a database or spreadsheets and, thus, at similar description AN, the operations over AN can be interpreted as transformations database or spreadsheets.

References.

1. Иванищев В.В. Алгоритмический базис для описания механизмов экономики. // Алгоритмические модели в автоматизации исследований и проектирования. М.: Наука, 1980. С.37-42.
2. Иванищев В.В., Марлей В.Е., Морозов В.П. Язык алгоритмических сетей. Л.:ЛНИВЦ АН СССР, 1984. 37с.
3. Иванищев В.В. Автоматизация моделирования потоковых систем. Л.:Наука, 1986. 142 с.
4. Марлей В.Е., Морозов В.П. Использование метода резолюций для планирования вычислений на алгоритмических сетях. // Методы и системы автоматизации в задачах науки и производства. М.: Наука,1986. С.23-34.
5. Иванищев В.В., Марлей В.Е., Морозов В.П. Система автомаизации моделирования САПФИР-ИСКРА. Основы построения системы. Л.:ЛИИАН, 1989. 63 с.
6. Марлей В.Е. Алгоритмические сети и параллельные граф-схемы алгоритмов. // Проблемы информационной технологии и интегральной автоматизации производства. Л.: Наука, 1989. С. 130-136.
7. Марлей В.Е. Входной язык системы САПФИР как язык программирования высокого уровня. // Международная конференция ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ-ПРОМЫШЛЕННОЕ ПРИМЕНЕНИЕ. Л.: ЛИИАН, 1990. С.240-242.
8. Иванищев В.В., Марлей В.Е., Морозов В.П. Диалоговая процедура автоматизации построения программного обеспечения модели на основе матричного представления ее сети. // Системы автоматизации в науке и производстве. М.: Наука, 1984. С.11-22.

FEATURES OF USE OF THE MULTIDIMENSION FUZZY QUALIFIERS IN DECISION-SUPPORT SYSTEM

Sergey Bagretsov and Alexander Yakovlev
Pushkin Higher Military College of radioelectronics of Air Defence
Department of Defence
Pushkin-9, St.-Petersburg 189620, Russia

Abstract. The basic components of intelligent decision-support system (DSS) are examined in this research. Based on the results of the analysis of knowledge organization in the control and management systems of complicated technical complexes the model of intelligent DSS as hierarchy of classification concepts (knowledge frames, in a general sense fuzzy) is synthesized, complete set of algorithms of classification, taking into account an illegibility of the initial information, it multy-dimensional, diverse character, is developed. Means of dialogue, reflecting hierarchical character of the analysis of situations and taking the decisions on system management by the operator, not being expert in the field of programming, are developed. They allow him to fill and to specify factografic the contents of a database, appropriate to frame model.

Keywords. Decision-support system, controlled system, expert system, classification, fuzzy logic, recognition of images, training sample.

1. State of a problem of automation of decision making. At the moment the problem of automation of the decision making with use of intelligent decision-support system (DSS) is considered most widely in the field of technical diagnosing. However traditional methods of diagnosing and the approaches to development of program-algorithmic maintenance have a number of disadvantages:

1. The accuracy of the received control and measuring information, and influencing destabilizing factors, which, as a rule, have non statistic character and are displayed multydimensional quantitative - qualitative images with complex structure is not taken into consideration.

2. Peculiarities of poly-purpose character of the accepted decisions are not taken into account.

3. The majority of the approaches is based on private models of the control of components of the controlled system. It results in significant expenses of time and program resources on development of the long-term software on traditional technology of automation.

4. Constant participation of the developers of the software in work the operators of controlled system (CS) for updating diagnostic data of formalized model of the control of functioning of CS is necessary.

In a sense the last disadvantage is absent in traditional expert systems (ES) in connection with simplicity and presentation of a procedure updating production of rules, and also due to presence of factor of trust.

However the use of production ES for the decision of problems of the control of the CS functioning and management has essential restrictions for the following reasons:

1. It is not obviously possible to take into account fuzzy, multydimensional, correlated character of the description of situations of the CS management, represented in the form of wholesome images.

2. The required efficiency of manipulations of rules quantity, which is necessary for the control of CS functioning in each moment of the collection of the control and measuring information is not provided. It is connected first of all with use of the primitive, not taking into account specificity of subject area, strategy selection of results.

3. Significant frequency of inquiries on a logic inference of results which are created by the received on an input ES results of measurements of parameters of CS functioning.

4. There is no real opportunity to use the information about time during a logic inference of results (i.e. representation of knowledge, connected to real time).

5. Complexity of creation of knowledge bases under the operational documentation.

Thus, the importance of transition from traditional systems of maintenance of individual activity of the operator in a control system to sophisticated ergatical complexes, integrating all functions of preparation, and in some situations the making a decision is obvious. So such integration of an automated working place of the operator should provide adaptation of its structure of interaction with the operator during functioning system and correlation with external environment.

2. Problem of development of model-algorithmic maintenance DSS. At the moment theoretical methods of designing of similar systems, ensuring only individual decisions of a problem and not making its coverage, are well developed. Methodologically they are based on principles of rigid logic and quasi-determination of interaction between CS and environment. This approach narrows down the area of application of the developed theoretical statements, as in the majority of cases inaccuracy of this interaction has fuzzy character. Fuzzy of interaction is determined on one hand by the fuzzy of the initial information, and on the other - it is a necessary condition of development of intelligent system, as its qualitative perfection, the updating of its structural and functional component is impossible without that degree of uncertainty, which is necessary for occurrence in the system of the new information and new strategy. Therefore, accepting as a whole conceptual rules of fundamental researches in this area, it is necessary to expand their theoretical and methodological base on the basis of the account and research, in the interests of development of DSS, fuzzy character of their interaction with external environment.

Taking into account the specified peculiarities, there is the necessity of formation for such systems the special technology for representation of knowledge, which should include:

- methods of an estimation of an opportunity of the management problems decision;
- methods of development of generalized structure and strategy of the problems decision in dynamic problem areas (i.e. problems of management in critical situations with an opportunity of a fast feedback) in view of multi-purpose character of the accepted decisions, multy-dimensional, diverse and fuzzy character of the description of situations, on which making a decision is required;
- methods revealing of the basic types of knowledge and mechanisms of their interaction during the decision of problems;
- methods of creation, on the basis of the developed model of knowledge, architecture DSS for the decision the problems of examined category;
- the technique of formation of knowledge bases DSS for problems of a examined category, allowing to ensure teleological purchase and formalization of knowledge of revealed types.

3. Principles of construction and specificity of realization offered by DSS. The basis of the examined DSS consists of set of original techniques of recognition multy-dimensional

fuzzy images [1], on the basis application of which the model-algorithmic description of processes of making decisions in various subject areas was realized.

Advantages of such approach are:

- opportunity of integral representation of analyzed situations;
- account of an illegibility of the initial data and character of the accepted decisions;
- account of set of the purposes of management;
- effective organization of the procedures of the searching and removing conflicting data in the knowledge bases and, also, procedures of inductive training of system.

In accordance with [2] DSS is formed on a basis of hierarchically connected with each other multy-dimensional fuzzy qualifiers, the structure of which reflects hierarchy of processing of the information in view of multi-purpose character of the accepted decisions. Each qualifier contains training sample (local base of knowledge about subject area), which representatively reflects the set of the accepted decisions at the given stage of computation. The training sample is a represented cortege of a kind $W=<X_i, \mu_j(X_i): i=1,N; j=1,K>$, where X_i - vector of parameters (in common kind quantitative - qualitative) describing object number i in training sample (TS); $\mu_j(X_i)$ - function of a fitting, determined on the set of the accepted decisions.

The work of the qualifier comes true in two stages. At the first stage fuzzy parameters separated surface in set of the relations system, submitted in TS are defined. They can be submitted as a cortege $V=< A_{\alpha b}, V_{j\alpha}: j=1,K; \alpha\in[0,1]>$, where $A_{\alpha\delta}$ - vector of weight factors of distances between the same parameters $x_\theta \in X$ classified object and object (b), accepted in set of objects TS for a beginning of coordinates; $V_{j\alpha}$ - vector of centers of classes; α - level of clearness of representation of knowledge, varied discretely in an interval [0,1]. At the second stage the calculated parameters of a scale of classification are used for classification of current situations, determined by a vector X, i.e. vector $\mu_j(X)$ (j=1,K) is defined.

Offered DSS represents an environment, to fill which expert owes. It is realized for work in Windows 3.11, has all service functions and is intended for:

1. construction of hierarchy of processing of the information according to revealed structure of the decisions making;
2. formation training samples for all qualifiers of the hierarchy;
3. formation of the purposes' priorities of the accepted decisions;
4. computation of the classification scales' parameters of the hierarchy's qualifiers;
5. estimation of reliability data in training samples;
6. classification of the acting information.

The information about the structure of hierarchy, the training samples and the data for classification with results of classification are stored in four tables:

- "BASE_***" - for each qualifier contains centers of classes, weight factors, number of classes, number of parameters, brief descriptions of used parameters, and also meaning of parameters of base object of training sample;
- "NAME_***" - contains the complete descriptions of all parameters, which use qualifiers of hierarchy;
- "OV___***" - contains training samples for all qualifiers of hierarchy;
- "CL___***" - contains the data on classification, the results of classification are brought in as well.

All tables have a format ".dbf" and if necessary can easily be corrected. On a place of symbols "***" in names of the tables there is the name of hierarchy, which is set by the user.

DSS can work in two modes:

1. computation of parameters of classification scales (data of the tables "BASE_***", "NAME_***", "OV___***" are used);

2. classification (data of the tables "BASE_***" , "NAME_***", "CL___***" are used).

On the figure 1 a function chart DSS is submitted. As it is obvious from a drawing, DSS consists from following of five subsystems:

1. Subsystem of hierarchies' manipulation - carries out reading hierarchies, creation of the new hierarchies and removal of the earlier created hierarchies.

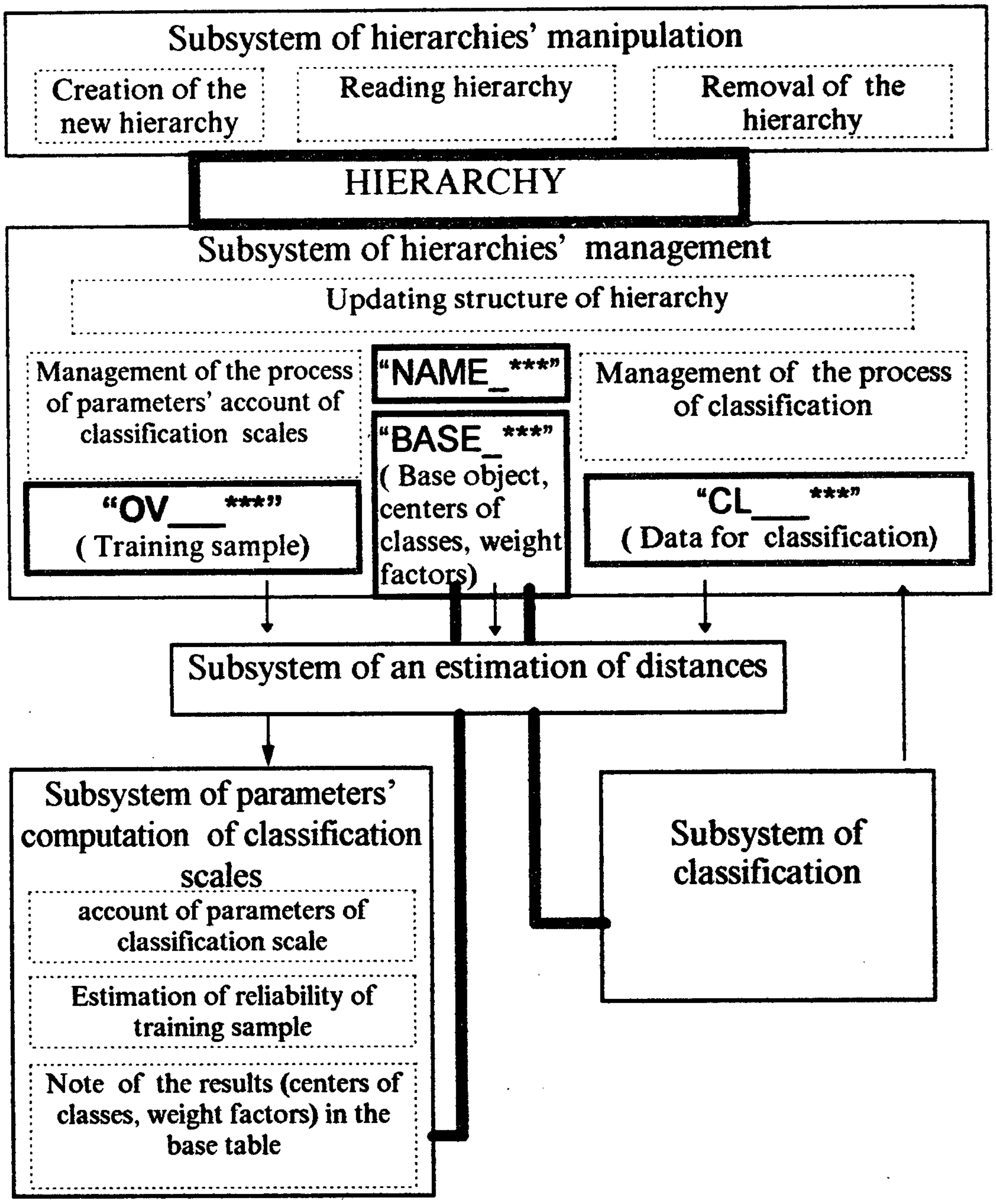


Fig. 1 A function chart DSS

2. Subsystem of hierarchies' management - modify the structure of hierarchy, manage processes of account of parameters of classification scales and make a classifications.

3. Subsystem of parameters' computation of classification scales.

4. Subsystem of classification.

5. The auxiliary subsystem of an estimation of distances - defines distance between meanings, submitted in the digital, fuzzy digital forms, sequences of numbers, binary sets, fuzzy sets, allocated parameters with probable or fuzzy distribution, line images, offers of natural language, precise and fuzzy logic designs.

4. Example of the realization of DSS. Here is an example of demonstration of work with DSS. Seventeen parameters {x1-x17} are given. Proceeding from results factor and claster analyses, it is known, that the relation between parameters have complex and nonlinear character. It is obvious, that the processing of the information cannot be made by means of the only one qualifier, and construction of hierarchy of qualifiers is required, the example of which is resulted on a figure 2. The given hierarchy has three levels and contains six qualifiers. Some of them will be described.

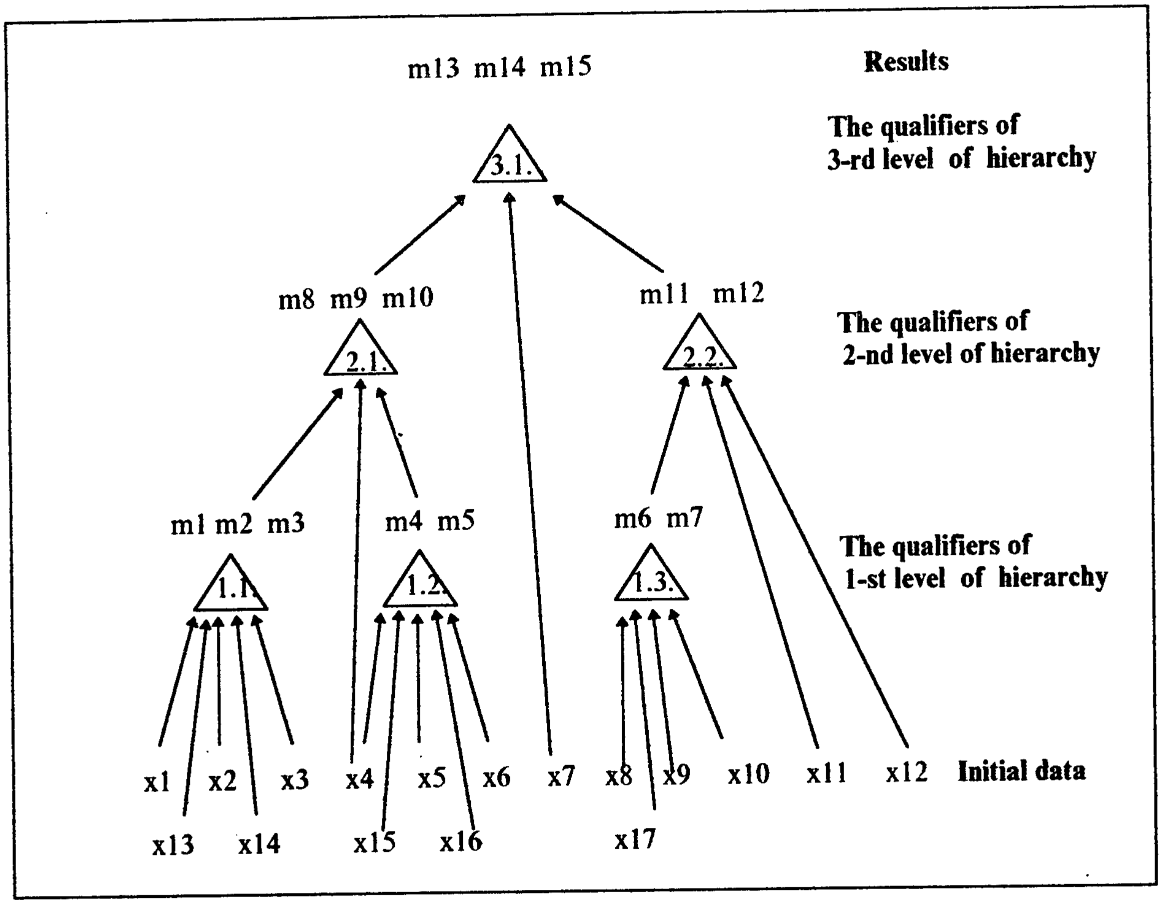


Fig. 2 An example of use of hierarchy of the qualifiers in DSS

At the input of the first qualifier of the first level of hierarchy (1.1) meanings of parameters {x1, x2, x3, x13, x14} come in; as a result of classification of the given group of parameters degrees of a fitting of object to three classes {m1, m2, m3} are defined. For example, in table 1 structure of training sample of the qualifier, realizing the first stage of making the decision on classification of flying objects, is resulted.

At the input of the second qualifier of the second level of hierarchy (2.2) meanings of parameters {x11, x12}, and also degree of a fitting of object to classes {m6, m7} (from the qualifier 1.3) come in. As a result of classification of the given group of parameters degrees of the fitting of object to two classes {m11, m12} are determined.

The third (last) level's qualifier of hierarchy (3.1) carries out the final classification of the object. At the input comes in meaning of a parameter {x7} and degree of a fitting of object to classes {m8, m9, m10} (from the qualifier 2.1) and {m11, m12} (from the qualifier 2.2); as a result of classification of the given group of parameters degrees of the fitting of object to three classes {m13, m14, m15} are defined, which is a result of classification.

Table 1 Example of filling of training sample for the qualifier 1.1

Meanings of parameters					degrees of the fitting of object to three classes		
speed of object	deviation	1-st corner of the direction	2-nd corner of the direction	3-rd corner of the direction	The plane	The helicopter	Other
x13	x14	x1	x2	x3	m1	m2	m3
800	0.99	70	15	12	0.97	0.05	0.00
750	0.97	82	17	56	0.91	0.10	0.00
600	0.95	68	9	14	0.85	0.10	0.05
500	0.86	80	5	25	0.81	0.15	0.11
250	0.67	145	12	2	0.58	0.45	0.13
200	0.68	154	16	34	0.42	0.68	0.23
150	0.56	132	7	10	0.33	0.87	0.35
75	0.45	0	0	25	0.21	0.45	0.51
20	0.05	9	5	90	0.10	0.31	0.72
5	0.05	15	0	4	0.01	0.23	0.89

Similarly by describing the other three qualifiers (1.2, 1.3, 2.1), the expert brings this information in DSS, then forms training sample for each of described qualifiers. On closing - up DSS calculates value of classification scale's parameters for each of qualifiers and makes an estimation of reliability of the putting in the information. After it DSS is ready for classification. In tables 2, 3, 4 results of computation of classification scale on the initial data of table 1 are submitted, and also example of classification of any flying object is resulted.

Table 2 Example of calculated parameters of a scale of classification (centers of classes) for the qualifier 1.1

Level of clearness of knowledge representation	center of classes m1 (the plane)	center of classes m2 (the helicopter)	center of classes m3 (other)
0.0	212.866	68.748	13.625
0.1	212.528	68.702	13.681
0.2	218.991	66.887	13.483
0.3	219.257	66.377	13.772
0.4	220.550	75.372	14.853
0.5	219.705	70.775	9.867
0.6	218.542	64.406	6.984
0.7	376.718	94.700	10.524
0.8	501.805	126.118	7.916
0.9	90.663	0.000	0.000

Table 3 Example of calculated parameters of the weight factors classification scale for the qualifier 1.1

Level of clearness of knowledge representation	weight factors for the parameter x13	weight factors for the parameter x14	weight factors for the parameter x1	weight factors for the parameter x2	weight factors for the parameter x3
0.0	0.300	0.550	0.050	0.050	0.050
0.1	0.300	0.550	0.050	0.050	0.050
0.2	0.300	0.550	0.050	0.050	0.050
0.3	0.300	0.550	0.050	0.050	0.050
0.4	0.300	0.550	0.050	0.050	0.050
0.5	0.300	0.550	0.050	0.050	0.050
0.6	0.300	0.550	0.050	0.050	0.050
0.7	0.525	0.306	0.056	0.056	0.056
0.8	0.700	0.075	0.075	0.075	0.075
0.9	0.100	0.600	0.100	0.100	0.100

Table 4 Example of classification of three flying objects by the qualifier 1.1

Initial data: meanings of parameters					Results: degrees of the fitting of object to three classes		
speed of object	deviation	1-st corner of the direction	2-nd corner of the direction	3-rd corner of the direction	The plane	The helicopter	Other
x13	x14	x1	x2	x3	m1	m2	m3
787	0.78	145	13	0	0.77	0.02	0.01
240	0.68	135	17	2	0.02	0.71	0.07
20	0.25	19	5	90	0.00	0.01	0.78

The presence of hierarchically allocated base of fuzzy knowledge has required development of the special apparatus of the control of completeness of the coordination of the preference relations of the accepted decisions and clearness of representation of situations at various levels of hierarchy, the principles of realization of which are stated in the research [4].

Conclusion. Use of principles of training classification in DSS allows to realize frame and production forms of reaction of DSS, to take into account their integral, multy-dimensional and diverse representation, to ensure the coordination of computation with structure of process of the decision making and to take into consideration their multi-purpose character. Within the above described approach, the problem of the search and removing conflicting data in the knowledge bases at all levels of hierarchy is effectively solved. Allocation of the procedures of parameters' account of classification scales and procedures of classification of situations itself allows to reduce time of search of the decisions up to levels, which admit use of DSS in a mode of real time. It can be used for the solution of a wide range of applied problems, when there is the element of uncertainty, and also there is a large number of not strongly correlated parameters.

References

1. Багрецов, С.А. (1991). Оценка знаний обучаемых в автоматизированных обучающих системах. Часть 1., ПВУРЭ, С.-Петербург.

2. Багрецов, С.А. (1996). Построение адаптивных организационных структур систем оперативного управления. Диссертация на соискание ученой степени доктора технических наук, ПВУРЭ, С.-Петербург.

3. Багрецов,С.А. (1995) Применение методов размытой классификации в диагностике сложных систем. Приборостроение, 9-10, с.72-76.

4. Багрецов, С.А., И.П. Горелов (1995). Последовательное агрегирование в оценке эффективности групповой деятельности операторов. Приборостроение,9-10, с.56-58.

CHAPTER VI:

COMPUTER LOGIC AND INTELLECTUAL PROCESSES

THIS CHAPTER INCLUDES PAPERS

PRESENTED AT THE CONFERENCE SESSION:

COMPUTER LOGIC AND INTELLECTUAL PROCESSES

Organized by: ***Prof. Vasily A. Lyubetsky,***

Prof. A. V. Chernavsky,

Prof. I. P. Lukashevitch.

ON SOME APPLICATIONS OF SEMANTIC EVALUATION METHOD

V. A. Lyubetsky
The Institute for Problems of Information Transmission
of Russian Academy of Sciences
Bolshoij Karetnij lane
Moscow 101447 RUSSIA
E-mail: Lyubetsk@ippi.ac.msk.su

Abstract. The problem is intended to solve is to develop a method of dealing with the "sense" (of judgments) suitable for the computer processing. In many cases the goal and the content of the computer treatment of information is to transform judgments in accordance with their "senses". The method consists in assigning to judgments certain elements of a specially chosen partially ordered set X. The element $[\![\varphi]\!]$ from X which corresponds to a judgment φ can be considered as "sense" of φ or "the degree of reliability of the judgment φ". The set X contains the greatest element $\top$ and the least element $\perp$. So a judgment φ is taken to be "true" (according to the assignment under consideration) if $[\![\varphi]\!] = \top$, and φ is "false" if $[\![\varphi]\!] = \perp$. We could hope for that some flexible systems of such assignments can be realized on a computer and this enables us to deal with "senses" more effectively. This idea arises at least in [2]. Such kind of assignments is called semantic evaluation with values in X. The method given in this paper can be mathematically considered as an effective variant of the sheaf theory on partially ordered sets. In this sense one can regard this method as belonging to Heyting-valued or Boolean-valued analysis.

Keywords. Description of object, classical and intuitionistic derivability, modelling intelligent systems, Heyting-valued analysis, transfer theorems.

In particular this approach showed to be useful in solution of the problem of finding unique (or the most individual) description of an object being given by a proof of its existence. It is also helpful for the task of transferring some classical propositions and estimates from field theory to rings of a special kind.

Here the former problem is considered as the problem of constructing "natural" derivability preserving translation from a classical theory to the corresponding nonclassical one. The question of complexity of the obtained description and about the efficiency of the arising algorithms for various fragments of classical set and type theories needs further investigation. The latter problem involves several questions which arise in coding theory. At last we have applied this technique to form a one intellectual model. As well as we have applied it to describe by some effective computer way a wide class of theories for which each formula is equivalent to the quantifier-free formula. See details in [3–5].

Let us recall that the set theory without the law of excluded middle preserves rich expressive possibilities for description of scenes, images, relations *etc.* of the classical set theory and at the same time has a lot of common features with an effective theory. For example, derivable formulas of the form $\forall x \exists y\, \varphi(x, y)$ often represent functions in a definable effective way.

Main results of this paper are as follows. Firstly, a wide class of formulas is constructed for which derivability in classical set theory implies intuitionistic derivability (see item 1). This result is a generalization of the well-known Friedman theorem on AE-arithmetical formulas [1].

Secondly, a sketch of one algorithm to modelling intellectual systems is described (see item 2). And the first result serves as a verification for this algorithm.

Thirdly, we present so-called transfer theorems for classical logic for the case of rings and ordered rings (see item 3). As an example we formulate generalized versions of Hilbert Nullstellensatz and Artin theorem on ordered fields for the class of regular functional rings, including the corresponding upper bounds. For example it is important for testing in coding theory.

1. A well known task is to construct a translation $\varphi \mapsto \varphi^0$ from a language with the symbol of equality $=$ and arbitrary functional and predicate symbols to itself such that classical deducibility of φ implies intuitionistic deducibility of φ^0, i.e. $T \vdash \varphi$, then $T^0 \vdash \varphi^0$, where T^0 is the intuitionistic counterpart of a theory T. Here we discuss the situation for the case T is Zermelo-Fraenkel set theory ZF (with ε-induction and collection or its suitable part) and T^0 is exactly the same theory without the law of excluded middle, noted as ZFI′. We give too a syntactical class of formulas φ, for which $\varphi^0 = \varphi$. H. Friedman stimulated the development of this direction by proving the basic result about AE-arithmetical formulas.

Let K be a countable structure. Let φ be a fi-formula, i.e. a formula such that the premise of any implication in it, first, has no $\forall$ and, second, $\exists$ is not under $\Rightarrow$. Let ψ be an AE-formula. Then $A \rightleftharpoons (\varphi \Rightarrow \psi)_K$ is a property of the structure K. Let A^0 be $(\forall x, y\, (P(x,y) \vee \neg P(x,y)) \wedge \ldots) \Rightarrow A$, where P,... are predicates of A. Our first result is:

Theorem 1. If ZF$\vdash$A then ZFI$\vdash$A^0.

The variable K may range over an absolute class of structures, defined by a set-theoretic formula (see [3]), or be free. The condition of countability of K can be loosened.

Proof. Assume the premise. What follows is a direct metamathematical description of the deduction in ZFI'.

Let $Z_2 \rightleftharpoons \{0, 1\}$; let $\mathcal{T}_2$ be the complete Heyting algebra of all the ideals a in Z_2 and A_2 - the complete Heyting algebra of all modal operators J in $\mathcal{T}_2$ [see 2]. There is an embedding of $\mathcal{T}_2$ into A_2 moving a to J_a, where $J_a(b) = a \bigvee b$. Let $\mathcal{B}_2 \rightleftharpoons \{J \in A_2 \,|\, (\neg\neg_{A_2})J = J\}$ be the complete Boolean algebra of all the regular elements of A_2. For any a from $\mathcal{T}_2$, J_a is a Boolean element of A_2, hence $J_a \in \mathcal{B}_2 \rightleftharpoons \{J \in A_2 \,|\, (\neg\neg_{A_2})J = J\}$.

We define next the evaluation

$[\![k = t]\!]_{\overline{K}} \rightleftharpoons \{0\} \cup \{x \,|\, x = 1,\, k = t\} \subset Z_2$, $[\![\cdot = \cdot]\!]_{\overline{K}} : K^2 \to \mathcal{T}_2$.

For any two terms s_1, s_2 define $[\![s_1(\overline{k}) = s_2(\overline{k})]\!]_{\overline{K}}$ as $[\![k = t]\!]_{\overline{K}}$, where $s_1(\overline{k}) = k$, $s_2(\overline{k}) = t$ are evaluated in K. In the same way define $[\![P(s_1, s_2)]\!]_{\overline{K}} \rightleftharpoons [\![P(s_1^0, s_2^0)]\!]_{\overline{K}}$, (where s_1^0, s_2^0 are the values of the terms in K). Extend the mapping $[\![\cdot = \cdot]\!]_{\overline{K}}$ from the set of all the terms with parameters from K to the set of all the formulas with parameters from K (without free variables). Using operations in $\mathcal{T}_2$ and $\mathcal{B}_2$ we have two valuations - $[\![\bullet]\!]_{\mathcal{T}_2}$ and $[\![\bullet]\!]_{\mathcal{B}_2}$ respectively.

As far as the condition

$$\forall k, t \in K \quad (k = t \vee k \neq t) \tag{$*_1$}$$

holds in $K \rightleftharpoons \omega$, $[\![s_1 = s_2]\!]_{\overline{K}}$ has only two values - $\{0\}$ and Z_2. Also $[\![P(s_1, s_2)]\!]_{\overline{K}} = \{0\}$ or $[\![P(s_1, s_2)]\!]_{\overline{K}} = Z_2$ because of the extra premise $P \vee \neg P$ in A^0, The same is true for $[\![s_1 = s_2]\!]_{\mathcal{B}_2}$ and $[\![P(s_1, s_2)]\!]_{\mathcal{B}_2}$, J_0, J_1 being its values.

Then, by induction on the construction of φ,

$$\varphi_{\overline{K}} \Leftrightarrow ([\![\varphi]\!]_{\mathcal{T}_2} = Z_2) \tag{1}$$

is proved for any formula φ.Thus, use of $[\![\bullet]\!]_{\mathcal{T}_2}$ is just a matter of presentation.

Let $V^{\mathcal{B}_2}$ be the Boolean-valued universum for the complete Boolean algebra $\mathcal{B}_2$. Let $(\bullet)^{\vee} : V \to V^{\mathcal{B}_2}$ be the usual embedding $x \mapsto x^{\vee} \rightleftharpoons \{y^{\vee} \mid y \in x\}_{-}$, where X_{-} is a function equal identically to $\top$ defined on X. It is known, that if ZF $\vdash \zeta$, then$[\![\zeta]\!]_{V^{\mathcal{B}_2}} = J_1$, and, in particular, $[\![f^{\vee} : (\omega^{\vee})^2 \to \omega^{\vee},\ P^{\vee} \subseteq (\omega^{\vee})^2 \wedge \ldots \Rightarrow \forall \overline{x}\ (\varphi \Rightarrow \psi)_{\omega^{\vee}, f^{\vee}, P^{\vee})}]\!] = J_1$,

$$[\![\varphi_{\omega^{\vee}}]\!]_{V^{\mathcal{B}_2}} \leq [\![\psi_{\omega^{\vee}}]\!]_{V^{\mathcal{B}_2}}. \tag{2}$$

Once we have (1), we need the premise of the Theorem 1 no more.

Suppose the condition

$$[\![k^{\vee} = t^{\vee}]\!]_{V^{\mathcal{B}_2}} \leq [\![k = t]\!]_{\overline{K}}. \tag{$*_2$}$$

Then induction on the construction of an arbitrary formula φ, we prove that

$$[\![\varphi\,(k_1, \ldots, k_n)]\!]_{\mathcal{B}_2} = [\![\varphi\,(k_1^{\vee}, \ldots, k_n^{\vee})_{\omega^{\vee}, f^{\vee}, P^{\vee}}]\!]_{V^{\mathcal{B}_2}}. \tag{3}$$

We prove also that:

a) for any fi-formula φ with parameters $\overline{k} = <k_1, \ldots, k_n> \in K$,

$$[\![\varphi(\overline{k})]\!]_{\mathcal{T}_2} \ \leq\ [\![\varphi(\overline{k})]\!]_{\mathcal{B}_2}. \tag{4a}$$

b) for any AE-formula ψ with parameters $\overline{k} = <k_1, \ldots, k_n> \in K$,

$$(a \leq [\![\psi(\overline{k})]\!]_{\mathcal{B}_2}) \Rightarrow (a \leq [\![\psi(\overline{k})]\!]_{\mathcal{T}_2}), \quad \forall a \in \mathcal{T}_2. \tag{4b}$$

Note, that all these auxiliary results are valid intuitionistically.

Finally, the proof is: suppose $f : \omega^2 \to \omega$, $P \subseteq \omega^2$ $\varphi_{\omega, f, P}(\overline{k})$. Suppose ($*_2$). Then, by (1), we have $[\![\varphi]\!]_{\mathcal{T}_2} = Z_2$ and by (4a) $[\![\varphi]\!]_{\mathcal{B}_2} = J_1$. By (3) $[\![\varphi(k_1^{\vee}, \ldots, k_n^{\vee})_{K^{\vee}}]\!]_{V^{\mathcal{B}_2}} = J_1$. From (2) $[\![\psi(k_1^{\vee}, \ldots, k_n^{\vee})_{K^{\vee}}]\!]_{V^{\mathcal{B}_2}} = J_1$, by (3) $[\![\psi(\overline{k})]\!]_{\mathcal{B}_2} = J_1$, and by (4b) $[\![\psi(\overline{k})]\!]_{\mathcal{T}_2} = Z_2$. Then $\psi_{\omega, f, P}(\overline{k})$ from (1). $\square$

The proof of Theorem 1 is based on the same technique as that of the proof of Theorem 2 (see below, item 3): for a structure K we define a non-standard representation K' in V^{Ω}, where V^{Ω} is the Heyting-valued universum for a suitable complete Heyting algebra Ω, such that $[\![K'$ is "simple"$]\!]_{\Omega} = \top$ (where $\top$ is the upper bound of Ω), while K is more "complicated".

For example, K is a strictly regular (ordered) functional ring (=f-ring) iff K' is a (linear) division ring; analogously, K is a projective ring iff K' is linear ordered, K is

a quasiregular f-ring iff K' is l-prime linear ordered, K is a projective f-ring without nilpotent elements iff K' is linear ordered without zero divisors, and so on.

The main point here is that a definite class of properties of K' is transferred as the class of properties of K; i.e. $[\![(K'$ is "simple"$) \Rightarrow \psi]\!]_\Omega = \top$ implies $K \models$ ("complicated" $\Rightarrow \psi$). $\square$

2. We can distinguish three components associated with the problem of modeling an intelligent system (IS) that is aiming at a goal. The description of the goal and of what "the goal is achieved" means is presented in item 2a. The description of the "memory," i.e.. the mechanism of the adequate (from the point of view of the given goal) reaction (excitation and inhibition, creation of an internal model of the situation) to the input information (the description of the "sensory" information, which enters the IS from the outside) is discussed in item 2b. The mechanism of choosing the plan of actions (for achieving the given goal) on the basis of the current input information and the mechanism of reflecting this information in the "memory" are discussed in item 2c.

Certainly, because of such a general statement of the problem and because of the absence of unambiguous experimental material for comparison, very different refinements of the notions are possible. However, it would be useful to compare these very refinements as well as to study the mathematical assertions and algorithms that lie at the foundation of these refinements.

The following item presents very schematically the framework for one possible refinement.

2a. On the reflex level, the plan of actions is automatically transformed into a chain of real actions. Here we shall not consider this transformation, and in this sense we will call the plan of actions the chain of actions.

Thus, the IS seeks a chain of actions $c_0, \ldots, c_l, \ldots$, that leads to the goal it desires. Here $c_0, \ldots, c_l, \ldots$ is a sequence of fixed letters, the set of which we shall denote by G. It is convenient to consider some more general situation when the chain of actions has the form $\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_i, \ldots$, where this chain represents a partition of the previous into segments, generally speaking, of unequal lengths. The IS assumes that the goal is achieved if the following system of conditions is met: $\vee_j \varphi_{1j}(\bar{c}_0, \bar{c}_1), \ldots, \vee_j \varphi_{ij}(\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{i-1}, \bar{c}_i)$, where $\varphi_{1j}, \ldots, \varphi_{ij}, \ldots$ are expressions (formulas) in a language that contains all the usual propositional symbols $\wedge$, $\vee$, $\neg$, $\Rightarrow$, $\Leftrightarrow$, and some fixed set of operational symbols $+$, $-$, $\cdot$,... and predicates. Denote $\varphi_i = \vee_{j=1}^{\infty} \varphi_{ij}$. This system has a natural interpretation: if the IS performs the actions $\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{i-1}$, then the next possible action $\bar{c}_i$, is subject to the restriction φ_i. Every condition φ_i, has the form $\vee_{j=1}^{\infty} \varphi_{ij}$ since: 1) it seems natural to describe a demand concerning an action $\bar{c}_i$, that continues the actions $\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{i-1}$ as a series of "homogeneous, similar in content" demands φ_{ij} (one of which is already sufficient); 2) as a whole, the system of conditions has the form of $\wedge_{i=1}^{\infty} \vee_{j=1}^{\infty} \varphi_{ij}$, where we recognize the conjunctive normal form, that is traditional in such problems.

A peculiarity of our situation is that we need to determine what is meant by "is satisfied," i.e., what sense (in connection with a given goal) we attach to the mentioned operations and predicates. To this end, the IS determines an interpretation in the set G of all these operations and predicates. In other words, it determines a full theory $\mathrm{Th}G = \{\varphi | G \models \varphi\}$ of the structure (algebra) G in the language mentioned, which is completed by the usual quantifiers. Intuitively, we consider $\mathrm{Th}G$ as a "coordinate system" that connects the symbols $c_0, \ldots, c_l, \ldots$ with the "reality that corresponds to the given

goal $\varphi_1, \ldots, \varphi_i \ldots$". In this sense, a goal is a system of conditions $\{\varphi_i\}$ plus a full theory ThG. This corresponds to a psychological observation: for solving a definite problem, a corresponding psychological reality is created, where the symbols $c_0, \ldots, c_l, \ldots$ acquire a definite content ThG that is associated with this problem. Thus, we want to find a structure G such that $G \models \wedge_{i=1}^{\infty} \forall \bar{x} \exists \bar{y} \vee_{j=1}^{\infty} \varphi_{ij}(\bar{x}, \bar{y})$, where $\bar{x}$ takes the place of the sequence $\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{i-1}$, and $\bar{y}$ takes place $\bar{c}_i$. Certainly, the original system of conditions in the structure G is satisfied in some strong universal sense: for every initial action $\bar{c}_0$, it is possible to find an action $\bar{c}_1$ such that $\varphi_1(\bar{c}_0, \bar{c}_1)$, and for every actions $\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{i-1}$, it is possible to find an action $\bar{c}_i$, such that $\varphi_i(\bar{c}_0, \bar{c}_1, \ldots, \bar{c}_{i-1}, \bar{c}_i)$. That is, in essence, the first condition upon the structure G consists in postulating the presence of functions $f_i(\bar{x}, \bar{y})$ which ensure the satisfaction of the conditions φ_i. Such a universality, clearly, corresponds to the intuitive ideas of an "image that corresponds to the realization of a given goal."

The second condition upon the structure G consists in the following. A full theory ThG is, in essence, an implicit (via its properties) description of operations and predicates in terms of which the conditions $\{\varphi_i\}$ are described. We wish to require a definite effectiveness from this description. Namely, we wish the truth in the structure G to be determined by its final part, i.e., ($G \models \varphi \Leftrightarrow \exists p(p$ is a finite set of basic formulas, $p \Vdash \varphi)$. Here, a formula is called basic if it is an atomic formula (or its negation) of the original language, and $\Vdash$ is the forcing predicate. In other words, the second condition upon the structure G is that it must be a generic structure in the sense to be explained in item 2c.

2b. We consider sheaves $\mathcal{F}(\cdot)$ that are defined on the full Heyting algebra Ω, which we identify with its Stone space $X(\Omega)$. For the sake of brevity, we will consider sheaves $\mathcal{F}(\cdot)$ that are defined on the topology $\mathcal{T}$ of some topological space X. All localizations $\{\mathcal{F}_x | x \in X\}$ of such a sheaf are algebras of the same structure as algebra G.

The current input information (the description of the outer world) T (at some nth instant in temporal development) excites a definite region in the space X of the sheaf $\mathcal{F}$: namely, an open set $\mathcal{O}(T, \mathcal{F}) = \{x \in X | \mathcal{F}_x \models T\}^0$, where Z^0 denotes the interior of the set Z. Here we may take into account not the "yes-no" excitation of the "cells," but the intensity of the excitations. Then the set $\mathcal{O}(T, \mathcal{F})$ is formed by distributions on the manifold X. However, now we consider a simpler situation.

If the excitation $\mathcal{O}(T, \mathcal{F})$ exceeds some threshold, e.g., in the sense that $\mu(\mathcal{O}(T, \mathcal{F})) > \lambda$, where $\mu(\cdot)$ is a fixed measure on X, and λ is, generally speaking, a varying number (or $\mathcal{O}(T, \mathcal{F})$ is massive in some topological sense), then the "module" $\mathcal{F}$ is actualized by the description T. (We may note a definite mutual subjection and interdependency of these modules; their role as elements, on the one hand, of the language, and on the other hand, of the "world".) The class $\mathcal{K}_T$ of sheaves is formed: $\mathcal{K}_T = \{\mathcal{F} | \mathcal{F}$ is actualized by the theory $T\}$. And we consider the theory Th$\mathcal{K}_T = \{\varphi | \forall \mathcal{F} \in \mathcal{K}_T(\mathcal{F}(X) \models \varphi)\}$ as the reflection of the input description T in the "consciousness of the IS." We will denote the condition in the braces by $\mathcal{K}_T \models \varphi$.

Speaking more precisely, we will define $\mathcal{K}_T$ in the following way: $\mathcal{K}_T = \{\mathcal{F} \in \mathcal{K} | \mathcal{F}$ is actualized by the theory $T\}$, where $\mathcal{K}$ is the class of "sheaves potentially prepared for actualization." The class $\mathcal{K} = \mathcal{K}_{n-1}$ reflects experience, learning that occurred prior to the nth instant (the moment of arrival of the input description T). The class $\mathcal{K}_{n-1}$ varies in time, i.e., $\mathcal{K}_n$ can differ from $\mathcal{K}_{n-1}$ in accordance with definite rules: a successful use of the theory $\mathbf{T}_n = \text{Th}\mathcal{K}_{T_n}$ prolongs the life of the class $\mathcal{K}_{n-1}$ (i.e., $\mathcal{K}_n = \mathcal{K}_{n-1}$); otherwise the class $\mathcal{K}_n$ extends the class $\mathcal{K}_{n-1}$. We do not discuss these rules here.

2c. The successive making of decisions (the choice of a chain of actions) proceeds in the following way. According to the input information T_n, the class $\mathcal{K}_{T_n}$ and the theory $\mathbf{T}_n = \mathrm{Th}\mathcal{K}_{T_n}$ are formed. A sequence of decisions $p_1 \subseteq \ldots \subseteq p_{n-1}$ is left from the previous iterations (and, by definition, $G = \cup_{n=1}^{\infty} p_n$). We choose the solution p_n from the conditions: $p_{n-1} \subseteq p_n$ and $\langle p_n, \mathbf{T}_n \rangle \Vdash \varphi_n$.

3. Now we formulate our Theorem 2 in a special case K being a commutative regular functional ring. Let $B(K)$ be the Boolean algebra of all the central idempotents of the ring K, $\mathcal{T}(K)$ - the algebra of all its ideals, and $\mathcal{B}$ is constructed over $\mathcal{T}$ in a usual way; $B \subseteq \mathcal{T} \subseteq \mathcal{B}$. Define $[\![\bullet]\!]_K$,$[\![\bullet]\!]_{\mathcal{T}}$ and $[\![\bullet]\!]_{\mathcal{B}}$ as in item 1, where $[\![s_1 \leq s_2]\!]_K \rightleftharpoons [\![k \leq t]\!]_K$ and k, t are the values of the terms s_1, s_2 in K. Then we will take $K' \rightleftharpoons \{P_k \,|\, k \in K\}_-$, $P_k(t^\vee) \rightleftharpoons [\![k = t]\!]_K$ as the representation of K in $V^{\mathcal{B}}$. Then assuming the condition

$$[\![k^\vee = t^\vee]\!]_{V^{\mathcal{B}(K)}} \leq [\![k = t]\!]_K, \quad \forall k, t \in K. \qquad (*_3)$$

We have

$$[\![\varphi(k_1, \ldots, k_n)]\!]_{\mathcal{B}(K)} = [\![(\varphi(P_{k_1}, \ldots, P_{k_n}))_{K'}]\!]_{V^{\mathcal{B}(K)}}. \qquad (5)$$

It is easy to calculate, that $[\![K'$ is a linear ordered field$]\!]_{\mathcal{T}} = \top$. Also, $[\![K'$ is a linear ordered field$]\!]_{\mathcal{B}} = J_1$, because it is a fi-formula.

We will define now an extension $\bar{K}$ of the ring K. There exists K'' in $V^{\mathcal{B}}$ such that $[\![K''$ is a real-closed field, $K' \subseteq K'']\!]_{V^{\mathcal{B}}} = J_1$.

Then

$$\overline{A} \rightleftharpoons (A'')^{\wedge_{\mathcal{B}(A)}} \rightleftharpoons \{g \in V^{\mathcal{B}(A)} \,|\, [\![g \in A'']\!]_{V^{\mathcal{B}}} = J_1\}.$$

Define $[\![\bullet]\!]_{\mathcal{B}(K),\overline{K}}$, as usual, $[\![f = g]\!]_{\mathcal{B}(K),\overline{K}} \rightleftharpoons [\![f = g]\!]_{V^{\mathcal{B}}}$ for any f,g$\in \bar{A}$, and similarly for $\leq$. The operations in $\bar{A}$ are induced by those in $(\bar{A})_-$. We prove that:

a) The ring $\bar{K}$ is an extension of the ring K with respect to the embedding $k \mapsto P_k$, including $\vee$ and $\wedge$;$[\![(\overline{K})_- = K'']\!]_{V^{\mathcal{B}}} = \top$; the evaluations $[\![\varphi]\!]_{\mathcal{B}(K),\overline{K}}$ and $[\![\varphi_{K''}]\!]_{V^{\mathcal{B}}}$ are identic, including operations $\vee$ and $\wedge$; the evaluation $[\![\bullet]\!]_{\mathcal{B}(K),\overline{K}}$ is equal to $[\![\bullet]\!]_{\mathcal{T}(K),K}$ on any atomic formula $s_1 = s_2$ and $s_1 \leq s_2$ (including operations $\vee$ and $\wedge$).

b) The structure $<\overline{K}, [\![\bullet]\!]_{\mathcal{B}(K),\overline{K}}>$ is a $\mathcal{B}$-orthocomplete real-closed commutative regular functional ring.

c) For any $\mathcal{B}$-orthocomplete real-closed commutative regular f-ring K_1, which is an extension of the ring K, there exists $K'' \in V^{\mathcal{B}}$, such that $[\![K''$ is a real-closed field, extending $K']\!]_{V^{\mathcal{B}}} = J_1$, and $K_1 = (K'')^{\wedge_{\mathcal{B}(K)}}$.

($\mathcal{B}$-orthocomplete means, by definition, that for any family $\{<b_\alpha, f_\alpha>\}$, $b_\alpha \in \mathcal{B}$, $f_\alpha \in K$ which satisfies the condition $b_\alpha \wedge b_\beta \leq [\![f_\alpha = f_\beta]\!]_{\overline{K}}, \forall \alpha, \beta$, there exists $f_0 \in K$ such that $b_\alpha \leq [\![f_0 = f_\alpha]\!], \forall \alpha$.)

Our original language will be now the "ordered field language": $=, +, -, \cdot, 0, 1, \ldots$, interpreted over K (here $x, y, \ldots$ range over K). To define the "input" formulas we extend this language with an additional sort of variables $\alpha, \ldots$ ranging over $\overline{K}$. In this language one defines: a P-formula is an atomic formula, or composed by using $\wedge, \forall x, \exists \alpha, \forall \alpha$ or a formula of the form $(\exists \alpha \varphi_1) \wedge (\forall \alpha(\varphi_1 \Rightarrow \varphi_2))$, where φ_1 and φ_2 are P-formulas; a weakly positive formula is an atomic formula, or composed by using $\wedge, \vee, \exists x, \forall x, \exists \alpha, \forall \alpha$ or a

formula of the form $(\exists\alpha\varphi_1) \wedge (\forall\alpha(\varphi_1 \Rightarrow \varphi_2)$, where φ_1 is a P-formula and φ_2 is weakly-positive; an "input" formula φ is a weakly positive formula, or of the form φ' (where φ is a fi-formula), or a weakly positive fi-formula in the original language, or constructed by using $\wedge, \vee, \exists x, \forall x, \exists\alpha, \forall\alpha$. The formula φ^0 is coming from φ by omitting all symbols $(\ )'$.

The two auxiliary facts are:

a) If φ is weakly-positive, then $(\varphi_K) \Rightarrow ([\![\varphi]\!]_{\mathcal{T}(K)} = \top)$

b) If ψ is weakly Horn, then $([\![\varphi]\!]_{\mathcal{T}(K)} = \top) \Rightarrow (\varphi_K)$.

Theorem 2. Let ψ be an AE-formula. If $\varphi^0 \Rightarrow \psi$ holds in the class of ordered fields K and their real-closed extensions $\overline{K}$, then $\varphi \Rightarrow \psi'$ holds in the class of commutative regular functional rings K and their $\mathcal{B}$-orthocomplete real-closed extensions $\overline{K}$.

In Theorem 2 one may take as $\overline{K}$ correspondingly only the real closure of a field K and the real closure of a ring K, namely $\overline{K} \rightleftharpoons (\overline{K'})^{\wedge\mathcal{B}}$.

For instance, assuming corresponding φ and ψ we get: for each regular functional ring K and a polynomial f over K, if $f \geq 0$ over $\overline{K}$, then f is a sum of squares with the same bounds as in the case of fields (in Artin theorem).

These bounds as well as the transformation of deductions (from Theorem 1) can be obtained by an explicit primitive recursive procedure.

The proof of Theorem 2 is quite similar to the proof of Theorem 1: provided φ is an input formula and $\varphi_{K,\overline{K}}$ we have $[\![(\varphi^0)_{K',(\overline{K})_-}]\!]_{V^{\mathcal{B}}} = J_1$. Then we have $[\![\psi_{K'}]\!]_{V^{\mathcal{B}}} = J_1$, because ZFC$\vdash(\varphi^0 \Rightarrow \psi)_{K,\overline{K}}$ for any ordered field K and its real-closed extension $\overline{K}$. Finally ψ'_K. $\square$

References

1. Friedman H. (1978) Classically and intuitionistically provable recursive functions. In Higher Set Theory. Lecture Notes in Mathematics, 669, Springer-Verlag, 21–27.

2. Fourman M. P., Scott D. S. (1979) Sheaves and logic. In Lecture Notes in Mathematics, 753. Berlin, Heidelberg, New York, Springer, 302–401.

3. Lyubetsky V. A. (1996) On some applications of semantic evaluation method. Problems of Information transmission of Russian Academy of Sciences, 32, No. 3, 128–143.

4. Velikhov Ye. P. and Tchernavsky A. V., eds. (1987) Intelligent Processes and Their Modeling [in Russian], Nauka, Moscow.

5. Lyubetsky V. A. (1988) On some applications of Heyting-valued analysis, II. In Lecture Notes in Computer Science, 417, 122–145.

A GEOMETRIC MODEL OF THE TRAJECTORY PLANNING

Alexey V.Chernavsky, Vladimir N.Karpushkin
Institute for the Information Transmission Problems RAS,
19, Bolshoy Karetny, Moscow, 1014407 Russia
E-mail: chernav@ippi.ac.msk.su

Abstract. A description of a neuronetwork model of the arm trajectory planning is described based on the introducing metrics given with the matrix elasticity coefficient.

Key words: intellect, sensory-motor control, arm trajectory, inverse kinematics problem, elasticity matrix coefficient, Riemann metrics, neuronetwork

The major task of intellect in the sensory-motor domain is the path planning on the basis of the geometric space-time presentation, which gives the possibility to replace optimising calculations with geometric constructions. In case of limb control it concerns the interrelations between four spaces, two exterior ones given by motor task and two their interior models. The exterior spaces are: the 3-dimensional space A in which the robot and the target of its movements are moving (the target space), and the space of the robot limb configurations D . Two interior spaces are their isomorphs: the 3-dimensional "visual space" $B \sim A$ and the control space $C \sim D$.

The interrelations between these four spaces may be presented with a simple cyclic scheme: the target point in A must have a reflection in B and cause the control signal to effectors (point in C), which must move the limb in such a position (point in D) that its end point ("end effector" or EE) coincided with the given target point.

The mapping $f\colon D \to A$ (and also $F\colon C \to B$) is defined "by arrangement of organ". The mapping $g\colon B \to C$ decides the "inverse kinematics problem" if the composition Fg is identity in B . To resolve this problem (to construct a g) one needs to bring into correspondence the frame systems in B and C, which is achieving with their ties to "natural" body coordinate system (upwards, forwards, rightwards). Because of movements of the body and its effectors the ties must be adjusted continuously.

The inverse kinematic problem has a solution based on the Hogan's proposal to use the minimisation of the potential elastic energy. If it is necessary to shift the EE through some vector v in A when the configuration of the limb is q , the new equilibrium position $q' = q+u$ must be chosen in such a way that it would be obtained under action of elastic forces if the EE were moved on the vector v , given the equilibrium q by some exterior force.

This suggestion is very commode from the mathematical point of view because it ties in an integrity three roles of one matrix Q . Firstly it is a matrix coefficient of elasticity in arm. Then the proposed solution coincides with the least squares method which provides the pseudoinverse matrix J^{-1} to the Jacobi matrix J for the mapping f:

$$J^{-1} = Q^{-1}J^{T}(JQ^{-1}J^{T})^{-1}$$